



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS JARDEL SILVA LOPES

**USO DE METAHEURÍSTICAS PARA AUXILIAR NO POSICIONAMENTO DE
UNIDADES BÁSICAS DE SAÚDE**

RUSSAS

2026

CARLOS JARDEL SILVA LOPES

USO DE METAHEURÍSTICAS PARA AUXILIAR NO POSICIONAMENTO DE UNIDADES
BÁSICAS DE SAÚDE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Universidade Federal do Ceará, como
requisito parcial à obtenção do grau de bacharel
em Ciência da Computação.

Orientador: Prof. Dr. Mayrton Dias de
Queiroz

RUSSAS

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

L851u Lopes, Carlos Jardel Silva.
USO DE METAHEURÍSTICAS PARA AUXILIAR NO POSICIONAMENTO DE UNIDADES
BÁSICAS DE SAÚDE / Carlos Jardel Silva Lopes. – 2026.
80 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas,
Curso de Ciência da Computação, Russas, 2026.
Orientação: Prof. Dr. Mayrton Dias de Queiroz.

1. Localização de facilidades. 2. Algoritmos Genéticos. 3. GRASP. I. Título.

CDD 005

CARLOS JARDEL SILVA LOPES

USO DE METAHEURÍSTICAS PARA AUXILIAR NO POSICIONAMENTO DE UNIDADES
BÁSICAS DE SAÚDE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Universidade Federal do Ceará, como
requisito parcial à obtenção do grau de bacharel
em Ciência da Computação.

Aprovada em: 27 de Janeiro de 2026

BANCA EXAMINADORA

Prof. Dr. Mayrton Dias de Queiroz (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Ms. José Robertty de Freitas Costa
Universidade Federal do Ceará (UFC)

Profa. Esp. Kátia Rocha de Almeida
Universidade Federal de Pernambuco (UFPE)

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe e pai, vocês foram minha motivação para suportar e superar todos obstáculos.

AGRADECIMENTOS

Ao Prof. Dr. Mayrton Dias de Queiroz por aceitar o desafio de me orientar em meu trabalho de conclusão de curso com tempo restrito e compartilhar todo seu conhecimento.

Aos minha mãe Ciene, meu pai Marcos e meus irmãos Aline e Joel, por toda educação e ensinamentos, reconheço todo o esforço e sacrifício para que eu tivesse uma formação superior.

A minha namorada Joana, presente comigo desde o primeiro dia de universidade, apoiando, ajudando, me distraíndo, um pilar para que eu pudesse chegar até aqui.

Aos amigos que fiz durante o curso, que foram muito importantes para de troca de conhecimentos e distrações para superar rotinas exaustas.

A Profa. Dra. Anna Beatriz dos Santos Marques, por todo apoio e ensinamentos repassados durante minha jornada de curso.

A associação Futuro da Criança, por todo material escolar e atividades de esporte, cultura, educação e capacitação profissional.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

“Se um dia tiver medo de voar, eu juro que abdicó dessas asas”

(Leall)

RESUMO

A Unidade Básica de Saúde (UBS) é o meio pelo qual a população tem acesso ao sistema de saúde, no entanto, é importante que elas sejam acessíveis à sociedade, possuindo uma boa localização. A correta localização dessas unidades é fundamental para garantir o acesso eficiente da população aos serviços básicos, especialmente em contextos marcados por limitações de recursos e infraestrutura. Dessa forma, surge o desafio de encontrar alternativas que permitam determinar a localização das UBSs. Diante deste cenário, o objetivo deste trabalho consiste em identificar uma estratégia capaz de determinar o posicionamento das Unidades Básicas de Saúde de uma forma que consiga atender o maior número de pessoas de uma região. Nesse trabalho foram selecionadas duas metaheurísticas: *Greedy Randomized Adaptive Search Procedure* (GRASP) e os Algoritmos Genéticos na resolução de um problema de localização de facilidades, com foco na maximização da cobertura de atendimentos em serviços de saúde pública. A definição da localização ideal das UBSs configura-se como um problema de otimização combinatória, pois envolve fatores como a distribuição geográfica da população, a limitação do número de unidades disponíveis e as distâncias entre os pontos de demanda. Devido à elevada complexidade computacional, a aplicação de métodos exatos torna-se inviável para instâncias de grande porte. Diante desse cenário, as metaheurísticas destacam-se como alternativas eficientes para obtenção de soluções de boa qualidade em tempo computacional reduzido, auxiliando a tomada de decisão. Os resultados obtidos indicaram que, para instâncias com menor número de UBSs, ambos os algoritmos apresentaram desempenhos semelhantes. À medida que o número de UBSs foi aumentado, o algoritmo GRASP obteve soluções de maior qualidade, alcançando ganhos percentuais superiores a 6% e 7% para quatro e seis UBSs, respectivamente. Em contrapartida, o Algoritmo Genético apresentou tempos de execução significativamente menores, com *speedups* superiores a 3,6 e 7,4, evidenciando a adequação de cada abordagem conforme as restrições de tempo e qualidade da solução.

Palavras-chave: localização de facilidades; algoritmos genéticos; GRASP.

ABSTRACT

The Basic Health Unit (UBS) is the means by which the population has access to the health system; however, it is important that they are accessible to society, possessing a good location. The correct location of these units is fundamental to guaranteeing the efficient access of the population to basic services, especially in contexts marked by limitations of resources and infrastructure. Thus, the challenge arises of finding alternatives that allow determining the location of the UBSs. Given this scenario, the objective of this work is to identify a strategy capable of determining the positioning of the Basic Health Units in a way that can serve the largest number of people in a region. In this work, two metaheuristics were selected: Greedy Randomized Adaptive Search Procedure (GRASP) and Genetic Algorithms in solving a facility location problem, focusing on maximizing the coverage of care in public health services. Defining the ideal location of primary healthcare units (UBSs) is a combinatorial optimization problem, as it involves factors such as the geographical distribution of the population, the limited number of available units, and the distances between demand points. Due to the high computational complexity, the application of exact methods becomes unfeasible for large instances. In this scenario, metaheuristics stand out as efficient alternatives for obtaining good quality solutions in reduced computational time, assisting in decision-making. The results obtained indicated that, for instances with a smaller number of UBSs, both algorithms showed similar performance. As the number of UBSs increased, the GRASP algorithm obtained higher quality solutions, achieving percentage gains greater than 6% and 7% for four and six UBSs, respectively. In contrast, the Genetic Algorithm presented significantly shorter execution times, with speedups greater than 3.6 and 7.4, demonstrating the suitability of each approach according to the time and solution quality constraints.

Keywords: facility location; genetic algorithms; GRASP.

LISTA DE FIGURAS

Figura 1 – Explicação do funcionamento de um Algoritmo Genético	26
Figura 2 – Passos realizados durante a metodologia deste trabalho	32
Figura 3 – Execução nº1 da instância t1000n0u2.txt no GRASP	40
Figura 4 – Execução nº1 da instância t1000n0u2.txt no AG	40

LISTA DE TABELAS

Tabela 1 – Organização do conjunto de instâncias	33
Tabela 2 – Comparação das médias entre GRASP e Algoritmo Genético (AG) para duas UBSs	41
Tabela 3 – Comparação das médias globais entre GRASP e AG para duas UBSs	42
Tabela 4 – Comparação das médias entre GRASP e AG para quatro UBSs	42
Tabela 5 – Comparação das médias globais entre GRASP e AG para quatro UBSs	43
Tabela 6 – Comparação das médias entre GRASP e AG para seis UBSs	43
Tabela 7 – Comparação das médias globais entre GRASP e AG para seis UBSs	44
Tabela 8 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u2.txt	50
Tabela 9 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u4.txt	51
Tabela 10 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u6.txt	52
Tabela 11 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u2.txt	53
Tabela 12 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u4.txt	54
Tabela 13 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u6.txt	55
Tabela 14 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u2.txt	56
Tabela 15 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u4.txt	57
Tabela 16 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u6.txt	58
Tabela 17 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u2.txt	59
Tabela 18 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u4.txt	60
Tabela 19 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u6.txt	61
Tabela 20 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u2.txt	62
Tabela 21 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u4.txt	63
Tabela 22 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u6.txt	64
Tabela 23 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u2.txt	65
Tabela 24 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u4.txt	66
Tabela 25 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u6.txt	67
Tabela 26 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u2.txt	68
Tabela 27 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u4.txt	69
Tabela 28 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u6.txt	70
Tabela 29 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u2.txt	71
Tabela 30 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u4.txt	72

Tabela 31 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u6.txt	73
Tabela 32 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u2.txt	74
Tabela 33 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u4.txt	75
Tabela 34 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u6.txt	76
Tabela 35 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u2.txt	77
Tabela 36 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u4.txt	78
Tabela 37 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u6.txt	79

LISTA DE QUADROS

Quadro 1 – Comparação entre os trabalhos relacionados.	31
--	----

LISTA DE ALGORITMOS

Algoritmo 1 – GRASP	24
Algoritmo 2 – FASE DE CONSTRUÇÃO DO GRASP	34
Algoritmo 3 – FASE DE BUSCA LOCAL DO GRASP	36
Algoritmo 4 – ALGORITMO GENETICO	37
Algoritmo 5 – Método da roleta	37

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
APS	Atenção Primária à Saúde
CTA	Centro Técnico Aeroespacial
ESF	Estratégia Saúde da Família
eSF	Equipe de Saúde da Família
FLP	Problemas de Localização de Facilidades
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
IEAv	Instituto de Estudos Avançados
INPS	Institutos Nacionais de Previdência Social
MCLP	<i>Maximal Covering Location Problem</i>
RCL	Lista Restrita de Candidatos
SCLP	<i>Set Covering Location Problem</i>
SUS	Sistema Único de Saúde
UBS	Unidade Básica de Saúde

LISTA DE SÍMBOLOS

α	Controle de equilíbrio do GRASP
g_{\max}	Ganho máximo
g_{\min}	Ganho mínimo
I	Conjunto de pontos de demanda
J	Conjunto de locais candidatos
w_i	Peso ou demanda associada ao ponto $i \in I$
d_{ij}	Distância entre o ponto de demanda i e o local candidato j
R	Raio máximo de cobertura
p	Número máximo de instalações a serem localizadas
x_j	Variável de decisão para instalação localizada no ponto candidato j
y_i	Variável de decisão para ponto de demanda coberto por instalação
N_i	Representa o conjunto de locais candidatos capazes de cobrir o ponto de demanda i

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Contextualização	18
1.2	Objetivo geral	19
1.2.1	<i>Objetivos específicos</i>	19
1.3	Justificativa	19
1.4	Organização do trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Atenção Primária à Saúde e Unidades Básicas de Saúde	21
2.2	Problemas de Localização de Facilidades	22
2.2.1	<i>O Maximal Covering Location Problem (MCLP)</i>	22
2.2.1.1	<i>Formulação Matemática</i>	23
2.3	Algoritmo GRASP	24
2.4	Algoritmos Genéticos	25
2.4.1	<i>Representação Cromossomial</i>	26
2.4.2	<i>Criação da População Inicial</i>	26
2.4.3	<i>Método de Avaliação</i>	27
2.4.4	<i>Seleção dos Pais</i>	27
2.4.5	<i>Crossover</i>	27
2.4.6	<i>Mutação</i>	27
3	TRABALHOS RELACIONADOS	29
3.1	Otimização da Localização das Unidades Básicas de Saúde Utilizando o Modelo p-Centro	29
3.2	Aplicação do método GRASP no problema de posicionamento de radares de vigilância	29
3.3	GRASP aplicado ao problema de alocação de antenas de transmissão	30
3.4	Algoritmo Genético aplicado ao problema de alocação/localização de facilidades.	30
3.5	Comparação entre os trabalhos relacionados	31
4	METODOLOGIA	32
4.1	Contextualização	32

4.2	Passo 1 - Obtenção do conjunto de instâncias	32
4.3	Passo 2 - Descrição conceitual dos algoritmos	33
4.3.1	<i>Visão Geral do Algoritmo GRASP</i>	33
4.3.1.1	<i>Fase de construção</i>	34
4.3.1.2	<i>Fase de busca local</i>	35
4.3.2	<i>Algoritmo Genético</i>	36
4.4	Passo 3 - Análise comparativa dos algoritmos	38
5	RESULTADOS	39
5.1	Resultados para cobertura de duas UBSs	41
5.2	Resultados para cobertura de quatro UBSs	42
5.3	Resultados para cobertura de seis UBSs	43
6	CONCLUSÕES E TRABALHOS FUTUROS	45
	REFERÊNCIAS	47
	GLOSSÁRIO	49
	APÊNDICE A – RESULTADOS OBTIDOS DOS EXPERIMENTOS	50

1 INTRODUÇÃO

Neste capítulo será apresentada a contextualização do trabalho, bem como o objetivo geral e os objetivos específicos, justificativa e a organização do trabalho.

1.1 Contextualização

Até a promulgação da Constituição Federal de 1988 a política da saúde pública brasileira caracterizava-se por um modelo restritivo e excludente, no qual o acesso aos serviços de saúde era limitado majoritariamente aos trabalhadores formais vinculados aos Institutos Nacionais de Previdência Social (INPS), assim deixando uma parcela da população à margem da assistência em saúde, aprofundando as desigualdades sociais e regionais no país. Então surgiu o Sistema Único de Saúde (SUS) instituído pela Constituição Federal de 1988 garantindo direito universal e igualitário aos serviços de saúde, uma das mais relevantes conquistas sociais do povo brasileiro no enfrentamento das desigualdades sociais (Romero, 2018).

Desde a implementação do novo modelo de saúde pública no Brasil, as UBS passaram a desempenhar papel central na organização da Atenção Primária à Saúde (APS). No âmbito desse modelo, a Estratégia Saúde da Família (ESF) destaca-se como principal ordenadora do cuidado, sendo capaz de resolver aproximadamente 85% dos problemas de saúde da comunidade, conforme diretrizes do Ministério da Saúde (Brasil, 2017). As UBSs são fundamentais no enfrentamento e na prevenção de epidemias, além de atuarem de forma contínua em campanhas de vacinação, programas de controle e acompanhamento de doenças crônicas e ações de saúde bucal. Atualmente, o país conta com mais de 44 mil Unidades Básicas de Saúde distribuídas em todo o território nacional, configurando-se como a principal porta de entrada do SUS (Brasil, 2025).

Cada UBS deve contar, no mínimo, com uma Equipe de Saúde da Família (eSF), sendo responsável pela oferta de atenção integral, contínua e resolutiva à saúde de uma população adstrita, estimada entre aproximadamente 2.400 e 4.000 pessoas por equipe. Esse modelo tem como finalidade fortalecer os vínculos de compromisso e de corresponsabilidade entre os profissionais de saúde e a comunidade atendida, promovendo maior efetividade das ações em saúde. Diante desse contexto, a implantação de novas UBSs requer especial atenção à escolha de sua localização, de modo a ampliar o acesso e alcançar o maior número possível de indivíduos dentro de sua área de abrangência. Ressalta-se, entretanto, que a definição do local deve ser

precedida da realização de estudos técnicos e ambientais, a fim de avaliar os impactos decorrentes da construção e assegurar a sustentabilidade da intervenção (Brasil, 2017).

Apesar da relevância das UBSs para a organização da Atenção Primária, a definição de sua localização espacial representa um desafio significativo para a gestão pública, especialmente em contextos de restrição orçamentária e desigual distribuição populacional. A escolha inadequada do local pode comprometer o acesso da população aos serviços de saúde, reduzir a efetividade das equipes e ampliar desigualdades territoriais. Nesse sentido, torna-se necessário o emprego de ferramentas analíticas e métodos quantitativos capazes de apoiar o processo decisório, considerando simultaneamente critérios geográficos, populacionais e de cobertura dos serviços de saúde.

1.2 Objetivo geral

Diante desse cenário, o objetivo geral deste trabalho consiste em identificar um algoritmo capaz de posicionar as Unidades Básicas de Saúde visando maximizar a cobertura populacional dentro de um raio de atendimento previamente estabelecido.

1.2.1 Objetivos específicos

Assim, é possível destacar os seguintes objetivos específicos:

- Obter os principais trabalhos relacionados a essa pesquisa;
- Identificar um algoritmo capaz de posicionar as unidades de saúde;
- Obter uma análise comparativa com base no conjunto de instâncias adotadas.

1.3 Justificativa

A adoção de algoritmos de otimização tem como objetivo proporcionar uma solução eficiente e computacionalmente viável para problemas de localização de facilidades em saúde, buscando atender ao maior número possível de indivíduos. Ao otimizar a distribuição espacial das unidades, o método contribui para a redução das distâncias de locomoção da população, diminuindo a necessidade de deslocamentos longos e, conseqüentemente, a demanda por meios de transporte, também podendo salvar vidas em atendimentos que necessitam de urgência. Além disso, essa organização territorial favorece a atuação dos agentes de saúde, facilitando o acesso aos atendimentos domiciliares e promovendo maior agilidade e efetividade na prestação dos

serviços. Dessa forma, o método apoia o planejamento territorial e a tomada de decisão no âmbito da gestão do SUS, contribuindo para uma alocação mais eficiente dos recursos públicos e para a ampliação do acesso aos serviços de saúde.

1.4 Organização do trabalho

No capítulo 1 deste trabalho foi apresentado a contextualização do problema, objetivos gerais e específicos e a sua justificativa.

No capítulo 2 foi abordado a fundamentação teórica da APS e UBS, problemas de Localização de Facilidades, *Maximal Covering Location Problem* (MCLP), GRASP e AG, assim construindo a base para a realização deste trabalho.

No terceiro capítulo contém os trabalhos que se relacionam com o propósito desta pesquisa, sendo eles "Otimização da Localização das Unidades Básicas de Saúde Utilizando o Modelo p-Centro", "Aplicação do método GRASP no problema de posicionamento de radares de vigilância", "GRASP aplicado ao problema aplicado ao problema de alocação de antenas de transmissão" e "Algoritmo Genético aplicado ao problema de alocação/localização de facilidades".

Durante o capítulo 4 foi discorrido sobre a metodologia e os passos utilizada para a realização do trabalho, e posteriormente nos capítulos 5 e 6 a exibição, discussão e conclusão dos resultados obtidos, também deixando possíveis caminhos para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta-se uma revisão teórica e tem como objetivo contextualizar o problema estudado e apresentar os principais conceitos e modelos relacionados à localização de instalações e à otimização combinatória. São discutidos fundamentos teóricos, modelos clássicos e abordagens heurísticas relevantes, os quais servem de base para a metodologia adotada e para a análise dos resultados apresentados nos capítulos subsequentes.

2.1 Atenção Primária à Saúde e Unidades Básicas de Saúde

A APS constitui o primeiro nível de atenção do SUS, sendo responsável pela organização do cuidado e pela coordenação das ações em saúde no território. Esse nível assistencial engloba ações de promoção da saúde, prevenção de agravos, diagnóstico, tratamento, reabilitação e manutenção da saúde, orientando-se por práticas contínuas e integradas voltadas às necessidades da população sem preconceitos ou privilégios de qualquer espécie (Brasil, 2017; Paim, 2020).

As UBS configuram-se como a principal estrutura física da APS, desempenhando o papel de porta de entrada preferencial do sistema de saúde. Nessas unidades são desenvolvidas ações individuais e coletivas, incluindo acompanhamento clínico, vigilância em saúde, campanhas de imunização e atividades educativas, com foco na continuidade do cuidado e no acompanhamento longitudinal dos usuários (Brasil, 2017).

No âmbito da Atenção Primária, destaca-se a ESF como modelo prioritário de organização dos serviços. Em seu modelo distribui diversas eSF, onde cada é composta por equipes multiprofissionais responsáveis por uma população adstrita em determinado território, atuando de forma contínua e integrada. Esse modelo favorece o fortalecimento do vínculo entre profissionais de saúde e comunidade, além de apresentar elevada capacidade de resolução dos problemas de saúde mais frequentes, conforme diretrizes do Ministério da Saúde (Brasil, 2017).

A territorialização constitui um elemento central na organização da APS, uma vez que orienta o planejamento das ações a partir das características sociodemográficas, epidemiológicas e geográficas da população atendida. A definição da área de abrangência das UBS e das equipes de Saúde da Família possibilita o mapeamento das necessidades locais e o direcionamento eficiente dos recursos disponíveis, contribuindo para a redução das desigualdades no acesso aos serviços de saúde (Paim, 2020).

Dessa forma, a APS, operacionalizada por meio das UBS e da ESF, constitui um eixo estruturante do SUS, sendo essencial para a promoção da saúde, a prevenção de doenças e a organização do sistema de saúde como um todo. A compreensão desses elementos é fundamental para o desenvolvimento de estratégias de planejamento e otimização da localização das UBSs, temática abordada neste trabalho.

2.2 Problemas de Localização de Facilidades

Os Problemas de Localização de Facilidades (FLP) constituem uma classe de problemas de otimização que busca determinar a posição mais adequada de instalações de serviço, de modo a atender uma demanda distribuída espacialmente, considerando critérios como custo, distância, cobertura e capacidade. Esses problemas são amplamente estudados em áreas como logística, planejamento urbano e saúde pública, especialmente quando o objetivo é maximizar o acesso da população a serviços essenciais (Daskin, 1995).

No estudo de ReVelle *et al.* (1970), são analisados modelos de localizações de propriedades públicas e privadas, evidenciando que os modelos de localização diferem quanto aos seus objetivos e restrições. No setor privado, o principal interesse está associado à maximização do lucro e minimização de custos, enquanto no setor público busca-se ampliar o atendimento à população ao menor custo possível para a sociedade. Nesse contexto, os problemas de localização no setor público são tradicionalmente classificados em duas categorias: serviços emergenciais e não emergenciais. Para o presente trabalho, a UBS enquadra-se como um serviço de natureza não emergencial, cujo objetivo central é garantir o atendimento integral e contínuo da população adscrita, promovendo maior acessibilidade aos serviços de atenção primária à saúde (Galvão *et al.*, 1999).

2.2.1 O Maximal Covering Location Problem (MCLP)

O MCLP foi formalmente introduzido por Church e ReVelle (1974) como uma extensão do problema de cobertura clássica, incorporando restrições realistas de recursos. Diferentemente do *Set Covering Location Problem* (SCLP), que exige a cobertura total da demanda, o MCLP parte do pressuposto de que, em muitos contextos reais, não é possível atender a todos os pontos de demanda devido a limitações orçamentárias ou estruturais. Dessa forma, o objetivo do modelo é maximizar a cobertura possível dentro dessas restrições.

No MCLP, considera-se um conjunto de pontos de demanda distribuídos em uma região e um conjunto de locais candidatos para a instalação de facilidades. Um ponto de demanda é considerado coberto se estiver localizado a uma distância menor ou igual a um raio de cobertura pré-estabelecido a partir de pelo menos uma instalação. O problema consiste em selecionar exatamente p locais candidatos de modo a maximizar a soma da demanda coberta.

2.2.1.1 Formulação Matemática

A formulação clássica do MCLP proposta por Church e ReVelle (1974) pode ser expressa como um modelo de programação inteira binária. Seja:

- I o conjunto de pontos de demanda;
- J o conjunto de locais candidatos;
- w_i o peso ou demanda associada ao ponto $i \in I$;
- d_{ij} a distância entre o ponto de demanda i e o local candidato j ;
- R o raio máximo de cobertura;
- p o número máximo de instalações a serem localizadas.

Definem-se as variáveis de decisão:

- $x_j = \begin{cases} 1, & \text{se uma instalação for localizada no ponto candidato } j, \\ 0, & \text{caso contrário;} \end{cases}$
- $y_i = \begin{cases} 1, & \text{se o ponto de demanda } i \text{ for coberto por ao menos uma instalação,} \\ 0, & \text{caso contrário.} \end{cases}$

O modelo matemático é dado por:

$$\max \sum_{i \in I} w_i y_i \quad (2.1)$$

sujeito a:

$$\sum_{j \in J} x_j = p \quad (2.2)$$

$$\sum_{j \in N_i} x_j \geq y_i, \quad \forall i \in I \quad (2.3)$$

$$x_j \in \{0, 1\}, \quad \forall j \in J \quad (2.4)$$

$$y_i \in \{0, 1\}, \quad \forall i \in I \quad (2.5)$$

onde $N_i = \{j \in J \mid d_{ij} \leq R\}$ representa o conjunto de locais candidatos capazes de cobrir o ponto de demanda i .

A função objetivo maximiza a soma ponderada da demanda coberta. A primeira restrição garante que exatamente p instalações sejam localizadas. A segunda restrição assegura que um ponto de demanda só pode ser considerado coberto se existir pelo menos uma instalação dentro do seu raio de cobertura.

2.3 Algoritmo GRASP

GRASP é uma metaheurística proposta por Feo e Resende (1995), amplamente utilizada para a resolução de problemas de otimização combinatória, especialmente aqueles classificados como NP-difíceis. O método baseia-se na execução iterativa de dois estágios principais: a fase de construção gulosa aleatorizada e a fase de busca local. Ao longo das iterações, diferentes soluções são geradas e avaliadas, sendo mantida aquela que apresenta o melhor valor da função objetivo conforme no Algoritmo 1.

Algoritmo 1: GRASP

```

início
  InputInstance()
  para critério de parada do GRASP não for satisfeito faça
    ConstructGreedyRandomizedSolution ( Solution )
    LocalSearch ( Solution )
    UpdateSolution ( Solution , BestSolutionFound )
  fim
rof
fim
return BestSolutionFound

```

Fonte: Feo e Resende (1995).

Na fase de construção, o GRASP emprega um critério guloso para orientar a seleção dos elementos que compõem a solução. Entretanto, diferentemente de métodos puramente gulosos, incorpora-se um componente de aleatoriedade por meio da Lista Restrita de Candidatos

(RCL). Essa lista é formada por soluções parciais de boa qualidade, definidas a partir de um parâmetro de controle denominado alfa, que regula o equilíbrio entre intensificação e diversificação do processo construtivo. Valores menores de α (alfa) tornam o método mais guloso, enquanto valores maiores aumentam o grau de aleatoriedade.

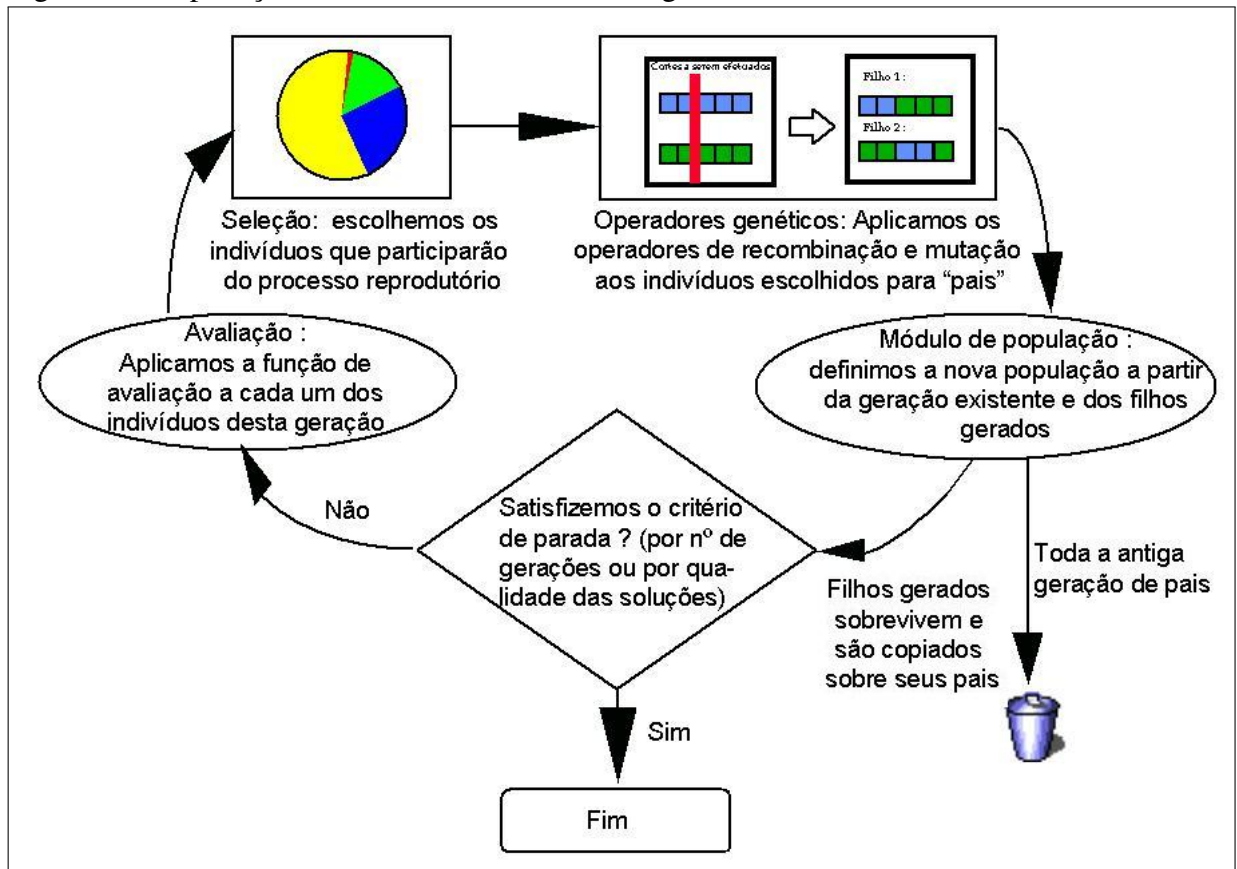
Após a construção da solução inicial, aplica-se uma etapa de busca local com o objetivo de aprimorar a solução obtida Queiroz e Silva (2024). Nessa etapa, são exploradas soluções vizinhas por meio de movimentos previamente definidos, tais como trocas, inserções ou remoções de elementos, dependendo da natureza do problema. O processo de busca local é interrompido quando nenhum vizinho capaz de melhorar a função objetivo é encontrado, caracterizando um ótimo local.

2.4 Algoritmos Genéticos

Com o objetivo de trazer a Teoria da Evolução por Seleção Natural, proposta por Darwin (1859), da biologia para os sistemas computacionais, John Holland desenvolveu os AG. Posteriormente, pesquisadores observaram que esses algoritmos apresentavam grande potencial para a resolução de problemas com alto grau de combinação, consolidando-os como uma importante técnica de otimização (Queiroz, 2018).

Para a resolução de problemas, os Algoritmos Genéticos seguem uma sequência de etapas que simulam o processo de adaptação evolutiva, visando alcançar de forma eficiente o objetivo proposto. Segundo Linden (2012), essas etapas são: (1) inicialização da população de cromossomos; (2) avaliação individual de cada cromossomo da população gerada; (3) seleção dos indivíduos pais; (4) aplicação dos operadores de cruzamento (*crossover*) e mutação sobre os pais selecionados, gerando novos cromossomos para a próxima geração; (5) exclusão dos pais da população atual; (6) avaliação dos descendentes e sua inserção na nova população; e (7) verificação do critério de parada, que pode ser o número máximo de gerações ou a obtenção de um indivíduo que satisfaça os requisitos de desempenho, conforme ilustrado no algoritmo apresentado na Figura 1.

Figura 1 – Explicação do funcionamento de um Algoritmo Genético



Fonte: Linden (2012).

2.4.1 Representação Cromossomial

No Algoritmo Genético, a representação cromossomial possui grande importância, pois constitui uma forma simples e eficiente de traduzir um problema real em um tratamento viável para o computador. Em outras palavras, trata-se do mecanismo que permite estabelecer a conexão entre o mundo real e o algoritmo genético, com o objetivo de possibilitar a obtenção das melhores soluções para o problema em estudo (Linden, 2012).

2.4.2 Criação da População Inicial

A criação da população inicial é uma etapa fundamental em Algoritmos Genéticos, pois define o conjunto de soluções iniciais (Queiroz *et al.*, 2016). Os indivíduos dessa população são os cromossomos e geralmente são gerados de forma aleatória para garantir diversidade genética, uma vez que uma população bem distribuída aumenta a probabilidade de obtenção de soluções ótimas ou próximas disso (Timóteo, 2002).

2.4.3 Método de Avaliação

A fim de determinar a qualidade de um indivíduo em relação à solução de um problema em estudo, os Algoritmos Genéticos utilizam um método de avaliação. Esse método atribui uma pontuação a cada indivíduo presente no conjunto de soluções, refletindo o quão adequada é sua representação para o problema considerado. Posteriormente, essa pontuação é utilizada pelo operador de seleção de pais, influenciando diretamente a escolha dos indivíduos que participarão do processo de reprodução (Linden, 2012).

2.4.4 Seleção dos Pais

Inspirado no mecanismo de seleção natural, o método de seleção de pais atua de forma que os indivíduos com maior aptidão gerem mais descendentes, mas também permite que indivíduos menos aptos possam gerar filhos. Nessa etapa, é de suma importância que o operador priorize os indivíduos bem avaliados, sem, contudo, descartar aqueles com menor pontuação, evitando, assim, que a diversidade genética seja perdida nas futuras gerações (Linden, 2012).

2.4.5 Crossover

O operador de *crossover*, também denominado cruzamento, baseia-se nos princípios da recombinação genética observada nos organismos biológicos durante o processo de reprodução. Nesse contexto, os descendentes são formados a partir da combinação das informações genéticas de dois indivíduos parentais, herdando características de ambos. Nos Algoritmos Genéticos, esse operador desempenha papel fundamental na exploração do espaço de busca, permitindo a geração de novas soluções a partir de soluções já existentes. Nos algoritmos genéticos de natureza contínua, existem diferentes estratégias de cruzamento, cuja aplicação depende diretamente do tipo de codificação e da representação adotadas para os indivíduos (Queiroz, 2018).

2.4.6 Mutação

Biologicamente, a mutação é um evento relativamente raro de ocorrer e, na maioria das implementações de Algoritmos Genéticos, também apresenta baixa frequência, sendo geralmente aplicada em cerca de 2% dos casos. O processo de definição da taxa de mutação está relacionado ao problema em questão, ao tamanho da população, à forma de codificação, entre outros fatores. Apesar de sua baixa frequência, a mutação desempenha um papel fundamental no

processo evolutivo, pois possibilita a introdução de novas características na população, contribuindo para a exploração do espaço de busca e para a obtenção de soluções potenciais para o problema (Queiroz *et al.*, 2019).

3 TRABALHOS RELACIONADOS

O presente trabalho se diferencia e, ao mesmo tempo, se relaciona com Laurindo e Lopes (2020) e Marques *et al.* (2017) ao abordar o problema da localização de unidades de saúde sob uma perspectiva complementar. Enquanto o trabalho Laurindo e Lopes (2020) foca na minimização da distância máxima por meio de um modelo exato (p-Centro) e Marques *et al.* (2017) foca na utilização de Algoritmo Genético, este trabalho propõe a utilização das metaheurísticas GRASP e AG, com o objetivo de maximizar a cobertura populacional dentro de um raio de atendimento previamente definido. O relacionamento com os trabalhos de Brito *et al.* (2005) e Arroyo e Marques (2006) estão presente na utilização da meta-heurística GRASP para a resolução do problema de cobertura máxima.

3.1 Otimização da Localização das Unidades Básicas de Saúde Utilizando o Modelo p-Centro

O estudo realizado por Laurindo e Lopes (2020) teve como objetivo analisar a aplicação do modelo de localização p-Centro para a alocação de UBS em um bairro da cidade de Teresina, capital do estado do Piauí. Foram avaliados dois cenários distintos: o primeiro considerou a instalação de novas unidades com a realocação das unidades já existentes; o segundo analisou a instalação de novas unidades sem a realocação das unidades em funcionamento. Para a coleta e organização dos dados, os autores utilizaram conceitos da Teoria dos Grafos, os quais subsidiaram a aplicação do modelo de localização p-Centro. Como métrica de avaliação, o estudo adotou o custo de deslocamento do usuário mais distante em relação à unidade de saúde mais próxima, buscando minimizar a maior distância percorrida pelos usuários. O estudo mostrou que a UBS existente não estava em localização ótima, com distância máxima de cerca de 1,9 km até a demanda mais distante. A instalação de três UBSs reduziria essa distância para aproximadamente 0,8 km, configurando um cenário ótimo para o bairro.

3.2 Aplicação do método GRASP no problema de posicionamento de radares de vigilância

O trabalho de Brito *et al.* (2005) investiga a aplicação da meta-heurística GRASP na resolução do problema de posicionamento de radares de vigilância, modelado como um FLP com MCLP. Considerando um número previamente definido de radares terrestres, o objetivo consiste em determinar suas localizações ótimas de modo a maximizar a cobertura da área sob vigilância,

atendendo ao maior número possível de demandas dentro da região protegida. A abordagem proposta combina estratégias gulosas e aleatórias, seguidas de busca local, permitindo explorar eficientemente o espaço de soluções e obter resultados de alta qualidade para um problema de natureza combinatória e complexa. O estudo é desenvolvido no âmbito do grupo de pesquisa de Apoio à Decisão do Instituto de Estudos Avançados (IEAv)/Centro Técnico Aeroespacial (CTA), contribuindo para o avanço de métodos computacionais aplicados à tomada de decisão em sistemas de defesa e vigilância territorial. O trabalho demonstrou a viabilidade do uso do GRASP na otimização do posicionamento de radares, apresentando resultados promissores mesmo sem garantia de otimalidade.

3.3 GRASP aplicado ao problema de alocação de antenas de transmissão

O trabalho realizado por Arroyo e Marques (2006) trata do problema de posicionamento de antenas de telecomunicações em uma região urbana, considerando aplicações como radiodifusão, transmissão de sinais de TV e Internet via rádio. O objetivo central é maximizar a cobertura dos pontos de demanda utilizando o menor número possível de antenas, respeitando restrições relacionadas ao alcance de transmissão e à presença de obstáculos que interferem na propagação do sinal. Para a resolução do problema, é proposta uma heurística baseada no GRASP, que incorpora um método guloso na fase construtiva e uma estratégia de cruzamento de soluções para aprimorar os resultados obtidos na busca local. A eficácia da abordagem desenvolvida é avaliada por meio de testes computacionais em instâncias de grande porte, tanto geradas manualmente, com soluções conhecidas, quanto criadas aleatoriamente, demonstrando o bom desempenho da heurística em diferentes cenários. O trabalho comparou duas heurísticas baseadas em GRASP para o posicionamento de antenas, mostrando que a versão aprimorada (GRASP2) superou a versão básica (GRASP1), com resultados próximos do ótimo e melhorias significativas nos problemas testados.

3.4 Algoritmo Genético aplicado ao problema de alocação/localização de facilidades.

O estudo realizado por Marques *et al.* (2017) analisa a aplicação de Algoritmos Genéticos ao problema de localização e alocação de postos de saúde, com o objetivo de maximizar a cobertura dos pontos de demanda utilizando o menor número possível de unidades. Diferentemente de abordagens que fixam previamente a quantidade de facilidades, o método

permite que o próprio algoritmo determine o número de postos mais adequados, a partir dos critérios estabelecidos. Os resultados obtidos em testes computacionais, envolvendo instâncias de pequeno a grande porte, demonstram a eficácia do Algoritmo Genético, evidenciada pela rápida evolução da população e pela qualidade das soluções encontradas.

3.5 Comparação entre os trabalhos relacionados

No Quadro 1, é possível observar a relação entre os trabalhos alinhados a desta pesquisa, tanto com relação ao contexto de aplicação dos trabalhos, bem como o algoritmo adotado por cada um.

Quadro 1 – Comparação entre os trabalhos relacionados.

Trabalho	Contexto	Algoritmos
Laurindo e Lopes (2020)	Otimização da Localização das UBS	p-Centro
Brito <i>et al.</i> (2005)	Problema de posicionamento de radares de vigilância	GRASP
Arroyo e Marques (2006)	Problema de alocação de antenas de transmissão	GRASP
Marques <i>et al.</i> (2017)	Problema de alocação/localização de facilidades	AG
Este trabalho	Otimização da Localização das UBS	GRASP e AG

Fonte: Elaborado pelo autor (2026).

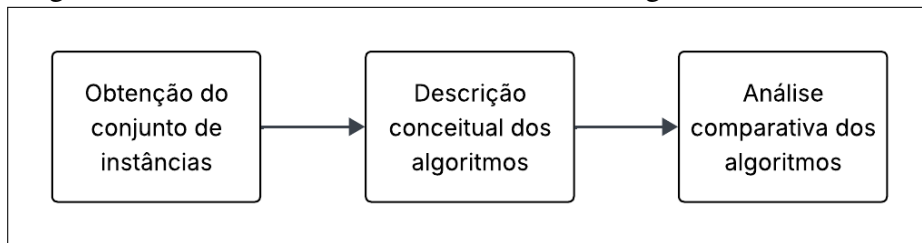
4 METODOLOGIA

Neste capítulo serão apresentados os passos realizados durante a metodologia deste trabalho.

4.1 Contextualização

Diante do objetivo deste trabalho, foi possível organizar a metodologia deste trabalho em 3 passos, conforme mostra a Figura 2. Adicionalmente, a descrição de cada passo será realizada na seções seguintes.

Figura 2 – Passos realizados durante a metodologia deste trabalho



Fonte: Elaborada pelo autor (2026).

4.2 Passo 1 - Obtenção do conjunto de instâncias

Com o intuito de observar o comportamento dos algoritmos e verificar o desempenho durante a sua execução, neste trabalho foram geradas 30 instâncias organizadas conforme mostra a Tabela 1. Cada instância utilizada possui:

Tamanho da região: valor correspondente a largura e comprimento da região, por exemplo,

1000 *km* de largura e 1000 *km* de comprimento;

Quantidade de UBS: número de USBs a serem posicionados na região;

Quantidade de pessoas: número correspondente a quantidade de pessoas;

Localização das pessoas: posição x e y de cada pessoa na região, correspondendo a latitude e longitude de uma pessoa.

Tabela 1 – Organização do conjunto de instâncias

Variação	Tamanho da região	Quantidade de UBS	Quantidade pessoas
Variação 0	1000	2	300
Variação 1	1000	2	300
Variação 2	1000	2	300
Variação 3	1000	2	300
Variação 4	1000	2	300
Variação 5	1000	2	300
Variação 6	1000	2	300
Variação 7	1000	2	300
Variação 8	1000	2	300
Variação 9	1000	2	300
Variação 0	1000	4	600
Variação 1	1000	4	600
Variação 2	1000	4	600
Variação 3	1000	4	600
Variação 4	1000	4	600
Variação 5	1000	4	600
Variação 6	1000	4	600
Variação 7	1000	4	600
Variação 8	1000	4	600
Variação 9	1000	4	600
Variação 0	1000	6	900
Variação 1	1000	6	900
Variação 2	1000	6	900
Variação 3	1000	6	900
Variação 4	1000	6	900
Variação 5	1000	6	900
Variação 6	1000	6	900
Variação 7	1000	6	900
Variação 8	1000	6	900
Variação 9	1000	6	900

Fonte: Elaborada pelo autor (2026).

4.3 Passo 2 - Descrição conceitual dos algoritmos

De posse das instâncias, neste passo foi iniciada a construção dos algoritmos a serem adotados no trabalho. Dessa forma, nas próximas subseções serão descritos os algoritmos adotados neste trabalho que são alinhados ao posicionamento das UBSs.

4.3.1 Visão Geral do Algoritmo GRASP

O algoritmo GRASP é um método iterativo composto por duas fases principais: (i) fase de construção gulosa randomizada, e (ii) fase de busca local. Essas duas fases são repetidas por um número pré-definido de iterações, sendo ao final selecionada a melhor solução encontrada ao longo do processo.

Inicialmente o algoritmo recebe como entrada um documento de texto correspondente

a instância, onde a primeira linha deve conter as dimensões da região, na segunda linha a quantidade de Unidades a serem instaladas, na terceira linha a quantidade de pessoas que habitam essa região e as demais linhas com as coordenadas geográficas de cada pessoa.

4.3.1.1 Fase de construção

Para a execução do Algoritmo 2, cria-se uma lista com as coordenadas de possíveis candidatos para o local de instalação das Unidades Básicas de Saúde na região, definida aleatoriamente dentro da região limite estabelecida pela a instância, sendo o tamanho fixo de dez candidatos.

Algoritmo 2: FASE DE CONSTRUÇÃO DO GRASP

Entrada: candidatosDisponiveis, pessoas, qtdUBS, raio, alpha

início

```

solucao = []
while tamanho(solucão) < qtdUBS AND candidatosDisponiveis != vazio do
  listaAvaliacoes = []
  for c de 0 até tamanhoCandidatos do
    ganhoCobertura = 0
    for pessoa in pessoas do
      if pessoaEstaCoberta(pessoa, solucao, raio) == false then
        if distanciaEuclidiana(candidato, pessoa) then
          | ganhoCobertura = ganhoCobertura + 1
        end
      end
    listaAvaliacoes.adicionar(candidato, ganhoCobertura)
  end
  ganhoMax = maior ganhoCobertura em listaAvaliacoes
  ganhoMin = menor ganhoCobertura em listaAvaliacoes
  limiteRCL = ganhoMax - alpha * (ganhoMax - ganhoMin)
  listaRCL = []
  for candidato in listaAvaliacoes do
    if candidato.ganho >= limiteRCL then
      | listaRCL.adicionar(candidato)
    end
  end
  candidatoAleatorio = aleatório da listaRCL
  solucao.adicionar(candidatoAleatorio)
  candidatosDisponiveis.remove(candidatoAleatorio)

```

end

return solucao

fim

Para a construção de ponto inicial para ser inserido na lista de possíveis candidatos, será explorado e analisado quais pessoas estão dentro do intervalo do raio. Nesse cálculo foi utilizada a distância Euclidiana para verificar se a população está contida no raio estabelecido pelas unidades. Para cada possível candidato é feito a coleta do ganho de cobertura e retirado o maior e o menor desta lista para a construção do nosso RCL feito pela seguinte equação:

$$RCL = g_{\max} - \alpha(g_{\max} - g_{\min}). \quad (4.1)$$

Para a equação foi utilizado o $\alpha = 0.30$ com o objetivo de garantir o controle entre gulosidade e aleatoriedade, então com a RCL definida tem-se os candidatos de possíveis soluções onde, é considerado como requisito o filtro de ganho de cobertura $> RCL$, e ao final foi selecionada uma quantidade de soluções de acordo com o limite de unidade estabelecido aleatoriamente dos candidatos que atenderam ao filtro.

4.3.1.2 *Fase de busca local*

Após a construção da solução inicial, aplica-se a fase de busca local conforme o Algoritmo 3 com o objetivo de aprimorar a solução obtida. Nessa etapa, são exploradas soluções vizinhas geradas a partir de pequenos deslocamentos espaciais nas posições das unidades instaladas.

A cada movimento, a nova solução é avaliada por meio da função objetivo, que contabiliza o número total de pessoas cobertas. Caso uma solução vizinha apresente melhoria em relação à solução corrente, ela passa a ser adotada como a nova solução base. O procedimento é repetido até que não seja possível encontrar melhorias adicionais, caracterizando a obtenção de um ótimo local.

Algoritmo 3: FASE DE BUSCA LOCAL DO GRASP

Entrada: solucao, pessoas, raio

início

```

melhorValor = calcularPessoasCobertas(pessoa, solucao, raio)
passo = 10
houveMelhora = true
while houveMelhora == true do
    houveMelhora = false
    for i in solucao do
        (x, y) = solucao[i]
        for dx in {-passo, 0, passo} do
            for dy in {-passo, 0, passo} do
                vizinho = solucao
                vizinho[i] = (x + dx, y + dy)
                valor = calcularPessoasCobertas(pessoas, solucao, raio)
                if valor > melhorValor then
                    solucao = vizinho
                    melhorValor = valor
                    houveMelhora = true
                end
            end
        end
    end
end
return solucao

```

fim

Fonte: Elaborado pelo autor (2026).

4.3.2 Algoritmo Genético

O Algoritmo Genético inicia com a criação da população inicial com indivíduos gerados aleatoriamente, com base nos limites do tamanho do território fornecido, representando possíveis soluções. Com a população definida inicia-se o *loop* até que a quantidade de gerações estabelecida seja atingida.

Dentro da estrutura de repetição, é calculada uma avaliação total com base na soma das avaliações individuais da lista de soluções candidatas. O método de avaliação adotado é a distância euclidiana entre as Unidades Básicas de Saúde e as pessoas. Essa soma global é posteriormente utilizada pelo operador de seleção de pais, por meio do método da roleta conforme o Algoritmo 4.

Algoritmo 4: ALGORITMO GENETICO

Entrada: tamanho, gerações, mutação

início

```

população = criarPopulação(tamanho)
para cont de 0 até gerações faça
  avaliar(população)
  para i de 0 até tamanho faça
    pai1 = selecionar()
    pai2 = selecionar()
    filho = reproduzir( pai1, pai2)
    novoFilho = mutar( filho, taxa)
    novaPopulação[i] = novoFilho
  fim
  moduloPopulação( população, novaPopulação)
  melhorFilho = selecionarMelhor( população)
fim
return melhorFilho
fim

```

Fonte: Elaborado pelo autor (2026).

A escolha dos pais foi realizada por meio da abordagem do método da roleta, amplamente utilizado em Algoritmos Genéticos, sendo o processo de seleção conduzido conforme o Algoritmo 5.

Algoritmo 5: Método da roleta

Entrada: avaliacao

início

```

posicao = numAleatorio .avaliacao
cont = 0
for i in populacao do
  cont = cont + populacao[i].getAvaliacao()
  if posicao <= cont then
    return i
  end
end
fim

```

Fonte: Elaborado pelo autor (2026).

Para cada indivíduo da população, realiza-se a escolha de dois pais utilizando o método da roleta, com base no filtro previamente definido. A partir dos pais selecionados, é gerado um descendente por meio da técnica de *crossover* de um ponto, formando, assim, uma nova população potencialmente mais adequada. Esta técnica consiste na seleção aleatória de um ponto ao longo do cromossomo dos indivíduos pais. Os genes anteriores ao ponto de corte são herdados do primeiro pai, enquanto os genes posteriores são herdados do segundo pai, resultando

na geração de um novo indivíduo. Esse processo visa combinar características genéticas distintas, aumentando a diversidade da população e auxiliando na convergência do algoritmo.

Em sequência ocorre a mutação com uma taxa de 0.1, assim explorando lugares com menor cobertura. Ao final da execução do algoritmo, é selecionado o indivíduo com maior potencial de cobertura.

4.4 Passo 3 - Análise comparativa dos algoritmos

De posse do planejamento dos algoritmos, em seguida os mesmos foram implementados e com o intuito de verificar seu comportamento foi realizado um experimento. No experimento, foi possível rodar as implementações dos algoritmos recebendo como entrada o conjunto de instâncias. Os resultados obtidos serão descritos no capítulo seguinte de Resultados. Adicionalmente, no Apêndice A, é possível observar os dados gerados após a execução dos experimentos.

5 RESULTADOS

Para a obtenção dos resultados, foram utilizadas dez variações de instâncias, cada uma com um mapa de 1000×1000 unidades de tamanho demarcando o território a ser analisado. Para cada variação, foram consideradas duas instâncias distintas: uma com a instalação de duas UBS, totalizando 300 indivíduos, outra com a instalação de quatro UBS, totalizando 600 indivíduos, e outra com a instalação de seis UBSs, totalizando 900 indivíduos.

Visto que os algoritmos possui chamada de funções pseudo-aleatórias, assim cada instância foi executada 30 vezes, com o objetivo de analisar o comportamento e a precisão dos algoritmos avaliados. Para cada execução foi recebida uma semente distinta, sendo elas formadas pelas casas decimais de π , ou seja, $s_1 = 141592$, $s_2 = 653589$, $s_3 = 793238$, ..., $s_{29} = 938521$ e $s_{30} = 105559$. Os experimentos realizados neste trabalho foram executados em uma máquina com as seguintes configurações: Processador Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (1.80 GHz), 12GB de RAM e Sistema Operacional Windows 11.

Cada instância é identificada pela nomenclatura “t1000n0u0.txt”, na qual “t” representa o tamanho da região, “n” indica o número da variação e “u” corresponde à quantidade de unidades. Todas as instâncias foram executadas utilizando a metaheurística GRASP e o Algoritmo Genético, adotando-se os mesmos parâmetros de configuração para ambos os métodos, a saber: 100 iterações e raio de cobertura igual a 200.

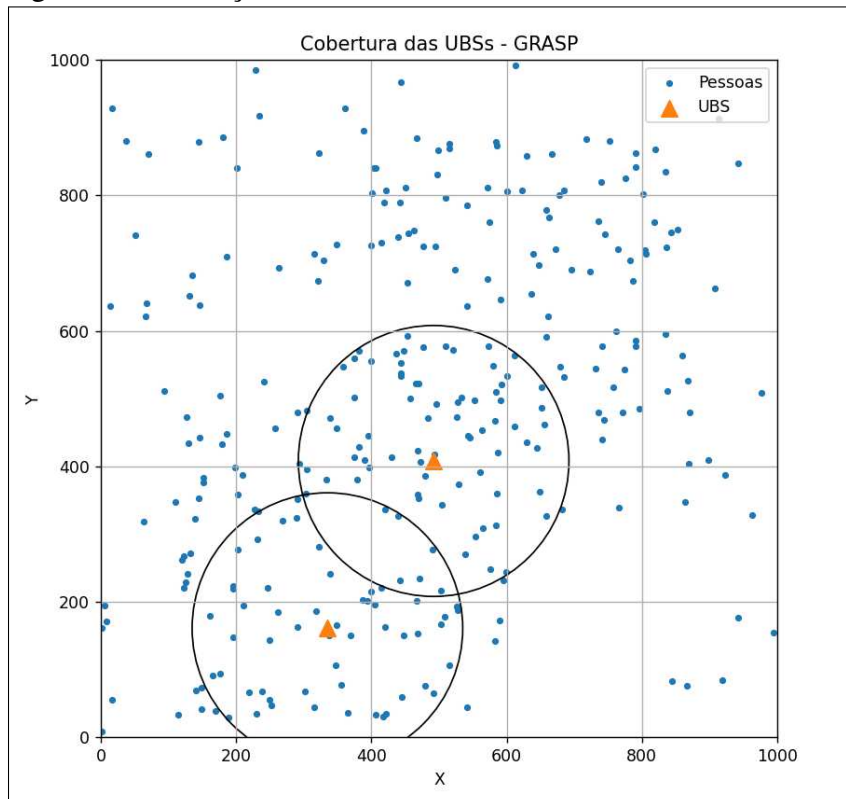
Cada execução do algoritmo apresenta como resultado a melhor cobertura encontrada, bem como os respectivos pontos de instalação das UBSs, conforme ilustrado nas Figuras 3 e 4 correspondentes. Esses resultados permitem observar, de forma clara, a distribuição espacial das unidades e o alcance obtido em cada cenário analisado.

Para análise dos dados coletados foram utilizados como métricas o ganho percentual da melhor solução em relação a menor solução realizado por meio da Equação 5.1 e a comparação do tempo de execução utilizando o fator de aceleração (*speedup*), definido na Equação 5.2.

$$\text{Ganho (\%)} = \frac{\text{Solução}_{\text{Maior}} - \text{Solução}_{\text{Menor}}}{\text{Solução}_{\text{Menor}}} \times 100 \quad (5.1)$$

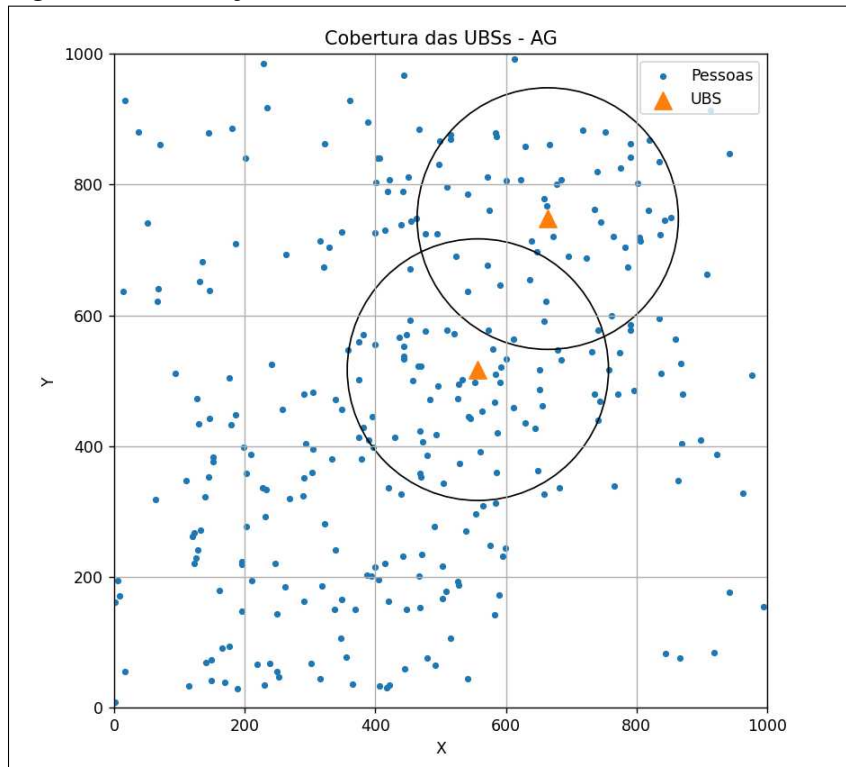
$$\text{Speedup} = \frac{\text{Tempo}_{\text{Maior}}}{\text{Tempo}_{\text{Menor}}} \quad (5.2)$$

Figura 3 – Execução nº1 da instância t1000n0u2.txt no GRASP



Fonte: Elaborada pelo autor (2026).

Figura 4 – Execução nº1 da instância t1000n0u2.txt no AG



Fonte: Elaborada pelo autor (2026).

5.1 Resultados para cobertura de duas UBSs

A Tabela 2 apresenta a comparação entre as médias das soluções obtidas pelos algoritmos GRASP e Algoritmo Genético para as instâncias com duas UBSs. Observa-se que ambos os métodos apresentaram desempenhos próximos em termos de qualidade de solução. Em algumas instâncias, como t1000n0u2.txt e t1000n2u2.txt, o Algoritmo Genético obteve valores médios ligeiramente superiores, enquanto em outras, como t1000n1u2.txt e t1000n9u2.txt, o GRASP apresentou melhores resultados.

Em relação ao tempo de execução, nota-se que os valores médios de ambos os algoritmos permanecem próximos, variando em torno de 2 a 3 segundos, sem diferenças expressivas que indiquem clara superioridade de um método sobre o outro para esse cenário.

Tabela 2 – Comparação das médias entre GRASP e AG para duas UBSs

Instância	GRASP		AG	
	Solução média	Tempo médio (s)	Solução média	Tempo médio (s)
t1000n0u2.txt	125	2.3134	127	2.3874
t1000n1u2.txt	180	2.5171	177	2.2732
t1000n2u2.txt	129	2.4301	130	2.3634
t1000n3u2.txt	137	2.4361	140	2.3227
t1000n4u2.txt	168	2.5066	169	2.2748
t1000n5u2.txt	160	2.9523	162	2.2827
t1000n6u2.txt	137	2.3201	141	2.3533
t1000n7u2.txt	150	2.2689	148	2.3242
t1000n8u2.txt	135	2.2747	137	2.3491
t1000n9u2.txt	167	2.5480	164	2.2898

Fonte: Elaborada pelo autor (2026).

Com base na Tabela 3, observa-se que, para o cenário com duas UBSs, o Algoritmo Genético apresentou uma média das soluções de 149.5, enquanto o GRASP obteve 148.8. Isso representa um ganho percentual aproximado de 0.47% em favor do Algoritmo Genético, indicando uma diferença pouco expressiva na qualidade das soluções encontradas.

Em relação ao tempo computacional, o Algoritmo Genético apresentou tempo médio de execução de 2.32 segundos, contra 2.46 segundos do GRASP, resultando em um *speedup* de aproximadamente 1.06, o que indica que o Algoritmo Genético foi cerca de 6% mais rápido. Apesar disso, essa diferença também se mostra reduzida, reforçando que, para instâncias com duas UBSs, ambos os algoritmos apresentam desempenho bastante semelhante, tanto em qualidade de solução quanto em eficiência computacional.

Tabela 3 – Comparação das médias globais entre GRASP e AG para duas UBSs

Algoritmo	Média das soluções	Média do tempo (s)
GRASP	148.8	2.46
Algoritmo Genético	149.5	2.32

Fonte: Elaborada pelo autor (2026).

5.2 Resultados para cobertura de quatro UBSs

A Tabela 4 apresenta os resultados médios para as instâncias com quatro UBSs. Diferentemente do cenário com duas UBSs, observa-se uma vantagem consistente do algoritmo GRASP em relação à qualidade das soluções. Em todas as instâncias analisadas, o GRASP obteve valores médios superiores aos do Algoritmo Genético, indicando maior capacidade de cobertura dos pontos de demanda quando o número de UBS é ampliado.

Entretanto, essa melhora na qualidade da solução ocorre à custa de um maior tempo computacional. Enquanto o GRASP apresenta tempos médios variando aproximadamente entre 21 e 28 segundos, o Algoritmo Genético mantém tempos significativamente menores, em torno de 6 a 7 segundos para todas as instâncias.

Tabela 4 – Comparação das médias entre GRASP e AG para quatro UBSs

Instância	GRASP		AG	
	Solução média	Tempo médio (s)	Solução média	Tempo médio (s)
t1000n0u4.txt	428	24.3053	397	6.7969
t1000n1u4.txt	439	28.3481	413	6.7466
t1000n2u4.txt	458	26.4005	432	6.6058
t1000n3u4.txt	410	27.0904	391	6.9723
t1000n4u4.txt	434	22.9522	410	6.7483
t1000n5u4.txt	396	21.0296	370	7.0217
t1000n6u4.txt	456	25.3696	411	6.7640
t1000n7u4.txt	447	27.1841	417	6.6870
t1000n8u4.txt	410	23.1395	389	6.9997
t1000n9u4.txt	413	22.1911	401	6.8181

Fonte: Elaborada pelo autor (2026).

De acordo com a Tabela 5, o algoritmo GRASP obteve uma média global das soluções igual a 429, enquanto o Algoritmo Genético alcançou 403. Esse resultado representa um ganho percentual aproximado de 6.45% em favor do GRASP, evidenciando sua maior eficiência na obtenção de soluções de melhor qualidade quando o número de UBS é ampliado.

Por outro lado, ao analisar o tempo computacional, observa-se que o Algoritmo Genético apresentou tempo médio de execução de 6.82 segundos, significativamente inferior ao

tempo médio do GRASP, que foi de 24.80 segundos. Dessa forma, o Algoritmo Genético obteve um *speedup* aproximado de 3.64 em relação ao GRASP, indicando que sua execução foi cerca de 3.6 vezes mais rápida. Esses resultados reforçam que, para instâncias com quatro UBSs, há um contraste claro entre qualidade de solução e tempo de execução, com o GRASP priorizando a qualidade e o Algoritmo Genético destacando-se pela eficiência computacional.

Tabela 5 – Comparação das médias globais entre GRASP e AG para quatro UBSs

Algoritmo	Média das soluções	Média do tempo (s)
GRASP	429	24.80
Algoritmo Genético	403	6.82

Fonte: Elaborada pelo autor (2026).

5.3 Resultados para cobertura de seis UBSs

A Tabela 6 apresenta a comparação entre as médias das soluções obtidas pelos algoritmos GRASP e Algoritmo Genético para as instâncias com seis UBSs. Observa-se que, em todas as instâncias analisadas, o GRASP obteve soluções médias superiores às do Algoritmo Genético, indicando maior capacidade de cobertura dos pontos de demanda à medida que o número de UBS aumenta.

Tabela 6 – Comparação das médias entre GRASP e AG para seis UBSs

Instância	GRASP		AG	
	Solução média	Tempo médio (s)	Solução média	Tempo médio (s)
t1000n0u6.txt	757	91.4543	715	12.7033
t1000n1u6.txt	730	96.3452	678	13.4045
t1000n2u6.txt	769	98.7420	707	12.8759
t1000n3u6.txt	757	98.6624	708	12.8444
t1000n4u6.txt	776	98.7144	729	12.4099
t1000n5u6.txt	768	90.9515	717	13.0079
t1000n6u6.txt	773	88.3292	720	12.7324
t1000n7u6.txt	749	97.4793	694	13.2064
t1000n8u6.txt	740	98.6821	692	13.2376
t1000n9u6.txt	768	99.9015	715	12.7762

Fonte: Elaborada pelo autor (2026).

Os valores médios das soluções do GRASP variaram entre 730 e 776, enquanto o Algoritmo Genético apresentou resultados entre 678 e 729. Esse comportamento evidencia que o GRASP mantém um desempenho consistente na qualidade das soluções mesmo em cenários mais complexos. Em contrapartida, os tempos de execução do GRASP foram significativamente

maiores, situando-se majoritariamente entre 88 e 100 segundos, enquanto o Algoritmo Genético manteve tempos médios próximos de 13 segundos em todas as instâncias.

A Tabela 7 consolida esses resultados ao apresentar as médias globais. O GRASP alcançou uma média das soluções de 759, enquanto o Algoritmo Genético obteve 707, o que representa um ganho percentual aproximado de 7.36% em favor do GRASP. Esse resultado reforça a superioridade do GRASP em termos de qualidade de solução quando o número de UBS é elevado.

Por outro lado, ao analisar o tempo computacional médio, observa-se que o Algoritmo Genético apresentou desempenho significativamente superior, com tempo médio de 12.92 segundos, contra 95.93 segundos do GRASP. Isso resulta em um *speedup* aproximado de 7.42, indicando que o Algoritmo Genético foi mais de sete vezes mais rápido que o GRASP nesse cenário.

Tabela 7 – Comparação das médias globais entre GRASP e AG para seis UBSs

Algoritmo	Média das soluções	Média do tempo (s)
GRASP	759	95.93
Algoritmo Genético	707	12.92

Fonte: Elaborada pelo autor (2026).

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho analisou a aplicação dos algoritmos GRASP e Algoritmo Genético ao problema de localização e alocação de Unidades Básicas de Saúde, com o objetivo de maximizar a cobertura populacional considerando diferentes cenários de complexidade. Foram avaliadas instâncias com duas, quatro e seis UBSs, permitindo analisar o impacto do aumento no número de unidades sobre a qualidade das soluções e o custo computacional dos algoritmos.

Os resultados obtidos para o cenário com duas UBSs indicaram que ambos os algoritmos apresentam desempenhos semelhantes, tanto em termos de qualidade das soluções quanto de tempo computacional. As diferenças observadas foram pouco expressivas, com ganhos percentuais e *speedup* reduzidos, sugerindo que, para instâncias menos complexas, qualquer uma das abordagens pode ser utilizada sem prejuízo significativo ao resultado final.

À medida que o número de UBSs aumenta, observa-se uma mudança clara no comportamento dos algoritmos. Para quatro e seis UBSs, o GRASP demonstrou maior capacidade de obtenção de soluções de melhor qualidade, apresentando ganhos percentuais progressivamente maiores em relação ao Algoritmo Genético. Esses resultados indicam que o GRASP é mais eficaz em explorar o espaço de busca quando o problema se torna mais complexo, resultando em maior cobertura da população.

Entretanto, essa melhoria na qualidade das soluções ocorre à custa de um aumento significativo no tempo computacional. O Algoritmo Genético destacou-se por apresentar tempos de execução substancialmente menores em todos os cenários analisados, alcançando *speedups* elevados em relação ao GRASP, especialmente nas instâncias com maior número de UBSs. Tal característica torna o Algoritmo Genético uma alternativa mais adequada quando o tempo de processamento é um fator crítico para a tomada de decisão.

Dessa forma, conclui-se que, para o conjunto de instâncias avaliadas, o Algoritmo GRASP mostrou-se mais adequado ao problema proposto, obtendo maior qualidade das soluções obtidas, porém em situações com restrições computacionais o AG demonstra boa eficiência pois tem menor custo computacional e uma boa qualidade de solução. Os resultados reforçam o potencial do uso de metaheurísticas no apoio ao planejamento territorial em saúde, contribuindo para decisões mais eficientes na alocação de recursos e ampliação do atendimento à população.

Como trabalhos futuros, sugere-se inicialmente a investigação de abordagens híbridas que combinem as características do GRASP e do Algoritmo Genético, visando explorar simultaneamente a qualidade das soluções e a eficiência computacional.

Também é relevante considerar a extensão do modelo para incorporar restrições e aspectos mais próximos da realidade, como a capacidade limitada das unidades de saúde, a possibilidade de geração de filas e variações temporais na demanda populacional. Por fim, a aplicação dos algoritmos a instâncias de maior porte e a dados reais pode fornecer uma avaliação mais abrangente do desempenho das abordagens propostas em contextos práticos de planejamento em saúde pública.

REFERÊNCIAS

- Arroyo, J. E. C.; MARQUES, T. B. Heurística grasp aplicada ao problema de alocação de antenas de transmissão. In: **Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional**. Goiânia, GO: SOBRAPO, 2006. Disponível em: <http://ws2.din.uem.br/~ademir/sbpo/sbpo2006/pdf/arq0199.pdf>. Acesso em: 18 jan. 2026.
- Brasil. **Manual de Estrutura Física das Unidades Básicas de Saúde**. Brasília: [s.n.], 2017. Disponível em: https://bvsmis.saude.gov.br/bvs/publicacoes/manual_estrutura_ubs.pdf. Acesso em: 03 dez. 2025.
- Brasil. **Censo das Unidades Básicas de Saúde**. Brasília: [s.n.], 2025. Disponível em: <https://www.gov.br/saude/pt-br/centrais-de-conteudo/apresentacoes/2025/censo-das-unidades-basicas-de-saude.pdf>. Acesso em: 03 dez. 2025.
- Brito, J. A. d. M.; MONTENEGRO, F. M. T.; BRITO, L. R. Grasp aplicado ao problema de posicionamento de radares. In: **Anais do XXXVII Simpósio Brasileiro de Pesquisa Operacional**. Gramado, RS: SOBRAPO, 2005.
- Church, R.; REVELLE, C. The maximal covering location problem. **Papers of the Regional Science Association**, v. 32, n. 1, p. 101–118, 1974.
- Darwin, C. **A origem das espécies**. Londres: John Murray, 1859.
- Daskin, M. S. **Network and Discrete Location: Models, Algorithms, and Applications**. Hoboken: John Wiley & Sons, 1995.
- Feo, T. A.; Resende, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, Springer, v. 6, n. 2, p. 109–133, 1995.
- Galvão, R. D.; NOBRE, F. F.; VASCONCELLOS, M. M. Modelos matemáticos de localização aplicados à organização espacial de unidades de saúde. **Revista de Saúde Pública**, v. 33, n. 4, p. 422–434, 1999. Disponível em: <https://www.scielo.br/j/rsp/a/jBDv8446myffDqZHjzSDZBS/?lang=pt>. Acesso em: 16 jan. 2026.
- Laurindo, L. E. C.; LOPES, I. J. Otimização da localização das unidades básicas de saúde utilizando o modelo p-centro: Um estudo de caso. In: **Anais do ENCOINFO – Congresso de Computação e Tecnologias da Informação**. Palmas, TO: CEULP/ULBRA, 2020. p. 104–113. ISSN 2447-0767. Disponível em: <https://ulbra-to.br/encoinfo/edicoes/2020/artigos/otimizacao-da-localizacao-das-unidades-basicas-de-saude-utilizando-o-modelo-p-centro-um-estudo-de-caso/>. Acesso em: 19 jan. 2026.
- Linden, R. **Algoritmos Genéticos**. 3. ed. S.l.: Brasport, 2012.
- Marques, T. B.; NETO, O. G. da S.; DINIZ, D. d. S. Algoritmo genético aplicado ao problema de alocação/localização de facilidades. In: SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Anais do Encontro Nacional de Computação dos Institutos Federais (ENCOMPIF)**. [S.l.], 2017. p. 256–259. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/99737715/9824-libre.pdf?1678614045>. Acesso em: 20 jan. 2026.
- Paim, J. S. **O que é o Sistema Único de Saúde**. Rio de Janeiro: Editora Fiocruz, 2020.

Queiroz, M.; MARTINS, M.; DANIEL, R. G.; LINO, N. Um estudo exploratório sobre o uso de algoritmos genéticos para o problema de eficiência energética em trens urbanos. **Anais do IX ENUCOMP**, 2016.

Queiroz, M. D. d. **SmartSubway: um sistema colaborativo para apoiar o estudo da eficiência energética em trens urbanos no contexto de cidades inteligentes**. Dissertação (Dissertação de Mestrado) — Universidade Federal da Paraíba, João Pessoa, PB, Brasil, 2018. Disponível em: <https://repositorio.ufpb.br/jspui/handle/123456789/18404>. Acesso em: 20 jan. 2026.

Queiroz, M. D. d.; SILVA, R. M. d. A. Framework para apoiar especialistas no problema de sequenciamento de contêiner em terminais portuários. **Revista Principia**, v. 61, n. 1, p. 31–57, jan. 2024. Disponível em: <https://periodicos.ifpb.edu.br/index.php/principia/article/view/6730>. Acesso em: 20 jan. 2026.

Queiroz, M. de; PALMEIRA, R.; MELO, F. de; DANIEL, R.; RIQUE Ícaro; GUIMARÃES, A.; MARTINS, M.; LINO, N. Um framework para apoiar especialistas no estudo da eficiência energética em trens urbanos. In: **Anais do XV Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2019. p. 32–39. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbsi/article/view/13886>. Acesso em: 21 jan. 2026.

Revelle, C. S.; MARKS, D. H.; LIEBMAN, J. C. An analysis of private and public sector location models. **Management Science**, v. 16, p. 692–707, 1970. Disponível em: <https://api.semanticscholar.org/CorpusID:154994170>. Acesso em: 07 jan. 2026.

Romero, L. C. P. O sistema único de saúde – um capítulo à parte. In: DANTAS, B. *et al.* (Ed.). **Sistema Único de Saúde**. Brasil: Editora, 2018. p. 67–88.

Timóteo, G. T. S. **Desenvolvimento de um algoritmo genético para a resolução do timetabling**. Trabalho de Conclusão de Curso — Universidade Federal de Lavras, Lavras-MG, 2002.

GLOSSÁRIO

AG Algoritmo Genético

APS Atenção Primária à Saúde

CTA Centro Técnico Aeroespacial

ESF Estratégia Saúde da Família

eSF Equipe de Saúde da Família

FLP Problemas de Localização de Facilidades

GRASP *Greedy Randomized Adaptive Search Procedure*

IEAv Instituto de Estudos Avançados

INPS Institutos Nacionais de Previdência Social

MCLP *Maximal Covering Location Problem*

RCL Lista Restrita de Candidatos

SCLP *Set Covering Location Problem*

SUS Sistema Único de Saúde

UBS Unidade Básica de Saúde

APÊNDICE A – RESULTADOS OBTIDOS DOS EXPERIMENTOS

Tabela 8 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n0u2.txt	132	2.7711	116	2.4130
Execução n° 2	t1000n0u2.txt	132	2.5643	130	2.4329
Execução n° 3	t1000n0u2.txt	131	2.3638	125	2.3241
Execução n° 4	t1000n0u2.txt	131	2.2578	128	2.3281
Execução n° 5	t1000n0u2.txt	121	2.0854	116	2.3533
Execução n° 6	t1000n0u2.txt	125	2.3131	132	2.3807
Execução n° 7	t1000n0u2.txt	108	1.5815	120	2.3655
Execução n° 8	t1000n0u2.txt	142	1.9017	126	2.3674
Execução n° 9	t1000n0u2.txt	125	2.3137	120	2.3257
Execução n° 10	t1000n0u2.txt	139	2.4048	139	2.3357
Execução n° 11	t1000n0u2.txt	121	1.9009	128	2.3397
Execução n° 12	t1000n0u2.txt	142	2.5047	125	2.3680
Execução n° 13	t1000n0u2.txt	134	3.3519	123	2.3988
Execução n° 14	t1000n0u2.txt	121	1.2036	129	2.3355
Execução n° 15	t1000n0u2.txt	113	2.4277	132	2.3817
Execução n° 16	t1000n0u2.txt	100	1.9437	127	2.3713
Execução n° 17	t1000n0u2.txt	143	2.4380	127	2.4205
Execução n° 18	t1000n0u2.txt	109	4.9607	131	2.4281
Execução n° 19	t1000n0u2.txt	125	2.7296	125	2.3800
Execução n° 20	t1000n0u2.txt	137	2.0208	131	2.4335
Execução n° 21	t1000n0u2.txt	105	2.8187	121	2.4346
Execução n° 22	t1000n0u2.txt	140	2.7283	127	2.4886
Execução n° 23	t1000n0u2.txt	124	1.7305	129	2.3661
Execução n° 24	t1000n0u2.txt	128	1.4879	128	2.4050
Execução n° 25	t1000n0u2.txt	135	2.4662	129	2.4913
Execução n° 26	t1000n0u2.txt	119	2.0225	127	2.4335
Execução n° 27	t1000n0u2.txt	120	2.7923	123	2.3930
Execução n° 28	t1000n0u2.txt	124	1.7321	130	2.4565
Execução n° 29	t1000n0u2.txt	128	1.9292	123	2.4053
Execução n° 30	t1000n0u2.txt	100	1.5341	128	2.5076

Fonte: Elaborada pelo autor (2026).

Tabela 9 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n0u4.txt	436	43.1991	410	6.7877
Execução nº 2	t1000n0u4.txt	427	34.0222	397	6.6182
Execução nº 3	t1000n0u4.txt	392	20.0341	385	6.9782
Execução nº 4	t1000n0u4.txt	421	24.3839	395	7.1361
Execução nº 5	t1000n0u4.txt	449	23.1677	387	6.7040
Execução nº 6	t1000n0u4.txt	443	23.2453	389	6.7734
Execução nº 7	t1000n0u4.txt	399	40.2522	403	7.0111
Execução nº 8	t1000n0u4.txt	461	35.4035	381	6.9531
Execução nº 9	t1000n0u4.txt	440	14.4373	422	6.6394
Execução nº 10	t1000n0u4.txt	449	19.1635	396	6.9437
Execução nº 11	t1000n0u4.txt	433	34.0232	414	6.7939
Execução nº 12	t1000n0u4.txt	433	40.1125	388	6.9122
Execução nº 13	t1000n0u4.txt	400	21.9149	397	6.8983
Execução nº 14	t1000n0u4.txt	456	25.8758	413	6.7092
Execução nº 15	t1000n0u4.txt	412	24.2268	390	6.9161
Execução nº 16	t1000n0u4.txt	412	41.3384	400	6.5733
Execução nº 17	t1000n0u4.txt	367	10.2612	401	6.6775
Execução nº 18	t1000n0u4.txt	430	25.9589	423	6.6488
Execução nº 19	t1000n0u4.txt	428	29.0080	402	6.8062
Execução nº 20	t1000n0u4.txt	331	15.5838	395	6.5974
Execução nº 21	t1000n0u4.txt	341	43.1327	393	6.9436
Execução nº 22	t1000n0u4.txt	429	16.3546	418	6.6934
Execução nº 23	t1000n0u4.txt	442	17.6998	435	6.5756
Execução nº 24	t1000n0u4.txt	437	16.6565	416	6.6576
Execução nº 25	t1000n0u4.txt	436	38.5490	416	6.9444
Execução nº 26	t1000n0u4.txt	418	23.8462	398	6.6248
Execução nº 27	t1000n0u4.txt	395	19.8226	394	6.8000
Execução nº 28	t1000n0u4.txt	396	18.0444	377	7.1627
Execução nº 29	t1000n0u4.txt	426	32.9034	374	6.9722
Execução nº 30	t1000n0u4.txt	451	30.6275	391	7.1448

Fonte: Elaborada pelo autor (2026).

Tabela 10 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n0u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº1	t1000n0u6.txt	765	66.5613	672	13.6046
Execução nº2	t1000n0u6.txt	786	49.4453	704	13.0617
Execução nº3	t1000n0u6.txt	783	95.6407	698	12.5906
Execução nº4	t1000n0u6.txt	787	40.7830	671	12.7772
Execução nº5	t1000n0u6.txt	758	109.7326	720	12.4203
Execução nº6	t1000n0u6.txt	719	139.6417	670	12.5715
Execução nº7	t1000n0u6.txt	772	120.9175	715	12.8266
Execução nº8	t1000n0u6.txt	753	58.3355	726	12.3014
Execução nº9	t1000n0u6.txt	770	58.5329	699	12.1642
Execução nº10	t1000n0u6.txt	745	89.5356	734	12.8715
Execução nº11	t1000n0u6.txt	754	35.3194	700	13.1215
Execução nº12	t1000n0u6.txt	762	106.9702	717	12.1018
Execução nº13	t1000n0u6.txt	777	102.8804	726	12.7275
Execução nº14	t1000n0u6.txt	773	153.8291	712	11.9868
Execução nº15	t1000n0u6.txt	760	131.4550	707	12.5711
Execução nº16	t1000n0u6.txt	725	74.3762	728	12.9578
Execução nº17	t1000n0u6.txt	742	82.3535	744	12.8627
Execução nº18	t1000n0u6.txt	762	127.3154	724	12.0376
Execução nº19	t1000n0u6.txt	675	95.0064	689	12.9989
Execução nº20	t1000n0u6.txt	749	64.8434	725	12.6791
Execução nº21	t1000n0u6.txt	775	111.5661	710	12.5394
Execução nº22	t1000n0u6.txt	735	90.8979	715	12.6509
Execução nº23	t1000n0u6.txt	740	83.0627	717	13.1485
Execução nº24	t1000n0u6.txt	750	81.8742	728	12.9457
Execução nº25	t1000n0u6.txt	738	92.0108	716	13.2145
Execução nº26	t1000n0u6.txt	756	110.4025	742	11.6364
Execução nº27	t1000n0u6.txt	773	54.7886	702	12.4771
Execução nº28	t1000n0u6.txt	751	103.4913	753	12.9336
Execução nº29	t1000n0u6.txt	786	124.3599	689	13.3689
Execução nº30	t1000n0u6.txt	721	70.2623	720	12.1178

Fonte: Elaborada pelo autor (2026).

Tabela 11 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n1u2.txt	176	3.8225	179	2.2602
Execução nº 2	t1000n1u2.txt	168	4.7786	177	2.3056
Execução nº 3	t1000n1u2.txt	175	2.2171	166	2.2636
Execução nº 4	t1000n1u2.txt	167	2.2517	175	2.3229
Execução nº 5	t1000n1u2.txt	175	1.3936	162	2.3410
Execução nº 6	t1000n1u2.txt	179	2.2808	167	2.3152
Execução nº 7	t1000n1u2.txt	183	3.9335	180	2.2435
Execução nº 8	t1000n1u2.txt	176	5.9865	182	2.2514
Execução nº 9	t1000n1u2.txt	185	1.4165	177	2.2680
Execução nº 10	t1000n1u2.txt	187	1.4407	172	2.3392
Execução nº 11	t1000n1u2.txt	181	2.2261	171	2.2706
Execução nº 12	t1000n1u2.txt	184	2.7902	176	2.2138
Execução nº 13	t1000n1u2.txt	190	5.8724	183	2.2339
Execução nº 14	t1000n1u2.txt	187	2.0511	182	2.2610
Execução nº 15	t1000n1u2.txt	139	2.0934	166	2.3056
Execução nº 16	t1000n1u2.txt	182	2.6140	171	2.3246
Execução nº 17	t1000n1u2.txt	163	8.3676	190	2.3204
Execução nº 18	t1000n1u2.txt	184	4.8704	182	2.3012
Execução nº 19	t1000n1u2.txt	155	2.1238	164	2.2097
Execução nº 20	t1000n1u2.txt	183	2.6808	159	2.3518
Execução nº 21	t1000n1u2.txt	153	1.0303	163	2.2759
Execução nº 22	t1000n1u2.txt	185	2.9032	192	2.1811
Execução nº 23	t1000n1u2.txt	169	3.3445	187	2.2902
Execução nº 24	t1000n1u2.txt	185	4.0631	183	2.2529
Execução nº 25	t1000n1u2.txt	181	1.3045	193	2.2883
Execução nº 26	t1000n1u2.txt	175	1.7841	180	2.1960
Execução nº 27	t1000n1u2.txt	162	2.4202	169	2.2445
Execução nº 28	t1000n1u2.txt	185	1.8922	177	2.3394
Execução nº 29	t1000n1u2.txt	129	4.9839	174	2.2329
Execução nº 30	t1000n1u2.txt	186	3.1100	179	2.3533

Fonte: Elaborada pelo autor (2026).

Tabela 12 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n1u4.txt	450	44.6224	423	6.8893
Execução nº 2	t1000n1u4.txt	422	24.9439	420	6.6369
Execução nº 3	t1000n1u4.txt	419	15.9436	409	6.6943
Execução nº 4	t1000n1u4.txt	427	13.1255	431	6.7014
Execução nº 5	t1000n1u4.txt	447	32.8295	435	6.5122
Execução nº 6	t1000n1u4.txt	444	26.1076	415	6.6095
Execução nº 7	t1000n1u4.txt	407	34.3445	403	6.6241
Execução nº 8	t1000n1u4.txt	452	26.6766	404	6.4958
Execução nº 9	t1000n1u4.txt	450	14.2727	388	6.8916
Execução nº 10	t1000n1u4.txt	397	16.8713	401	7.2777
Execução nº 11	t1000n1u4.txt	442	18.6382	422	6.7988
Execução nº 12	t1000n1u4.txt	460	41.8809	415	7.0678
Execução nº 13	t1000n1u4.txt	468	31.8827	397	7.0534
Execução nº 14	t1000n1u4.txt	448	29.8229	412	6.7516
Execução nº 15	t1000n1u4.txt	437	43.8906	409	7.2340
Execução nº 16	t1000n1u4.txt	436	40.6741	416	6.6085
Execução nº 17	t1000n1u4.txt	408	16.6049	414	6.7417
Execução nº 18	t1000n1u4.txt	442	32.5608	418	6.7045
Execução nº 19	t1000n1u4.txt	422	25.1102	411	6.7130
Execução nº 20	t1000n1u4.txt	418	19.5492	417	6.9716
Execução nº 21	t1000n1u4.txt	434	33.0544	427	6.6246
Execução nº 22	t1000n1u4.txt	442	14.4755	397	6.8191
Execução nº 23	t1000n1u4.txt	427	29.8453	417	6.7952
Execução nº 24	t1000n1u4.txt	445	24.7878	408	6.6999
Execução nº 25	t1000n1u4.txt	430	28.8169	397	7.3593
Execução nº 26	t1000n1u4.txt	444	27.8794	394	6.8778
Execução nº 27	t1000n1u4.txt	445	29.1695	441	6.6538
Execução nº 28	t1000n1u4.txt	454	29.8629	409	6.8692
Execução nº 29	t1000n1u4.txt	411	26.3798	396	7.0339
Execução nº 30	t1000n1u4.txt	414	37.9391	419	6.7043

Fonte: Elaborada pelo autor (2026).

Tabela 13 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n1u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº1	t1000n1u6.txt	752	120.9861	704	13.2286
Execução nº2	t1000n1u6.txt	740	95.1307	675	13.5121
Execução nº3	t1000n1u6.txt	679	101.3182	685	13.2170
Execução nº4	t1000n1u6.txt	692	77.9932	666	13.7492
Execução nº5	t1000n1u6.txt	743	75.5329	704	13.6858
Execução nº6	t1000n1u6.txt	749	102.2274	661	13.1331
Execução nº7	t1000n1u6.txt	701	109.1739	637	13.1827
Execução nº8	t1000n1u6.txt	755	89.3752	660	13.4332
Execução nº9	t1000n1u6.txt	753	48.5425	685	13.3078
Execução nº10	t1000n1u6.txt	751	99.6648	671	13.2076
Execução nº11	t1000n1u6.txt	721	91.0752	696	13.8235
Execução nº12	t1000n1u6.txt	766	100.0478	696	13.5539
Execução nº13	t1000n1u6.txt	764	104.0168	681	12.7907
Execução nº14	t1000n1u6.txt	729	63.4467	671	13.4524
Execução nº15	t1000n1u6.txt	678	102.8732	656	13.5251
Execução nº16	t1000n1u6.txt	715	97.5598	687	13.7182
Execução nº17	t1000n1u6.txt	713	132.0633	688	13.3758
Execução nº18	t1000n1u6.txt	710	103.2314	702	13.1503
Execução nº19	t1000n1u6.txt	722	73.3655	671	13.7177
Execução nº20	t1000n1u6.txt	676	58.3257	667	13.0858
Execução nº21	t1000n1u6.txt	670	143.5438	682	13.6559
Execução nº22	t1000n1u6.txt	738	87.3711	701	13.1817
Execução nº23	t1000n1u6.txt	720	81.4228	657	13.8736
Execução nº24	t1000n1u6.txt	754	69.4549	692	13.2153
Execução nº25	t1000n1u6.txt	716	78.7901	676	13.3268
Execução nº26	t1000n1u6.txt	758	113.7313	668	13.0557
Execução nº27	t1000n1u6.txt	757	92.6133	629	13.5426
Execução nº28	t1000n1u6.txt	731	104.2627	649	13.4366
Execução nº29	t1000n1u6.txt	747	147.3439	709	12.5925
Execução nº30	t1000n1u6.txt	667	83.8823	687	13.5754

Fonte: Elaborada pelo autor (2026).

Tabela 14 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n2u2.txt	126	1.3633	128	2.3593
Execução nº 2	t1000n2u2.txt	123	2.4591	121	2.3773
Execução nº 3	t1000n2u2.txt	135	1.9132	130	2.3073
Execução nº 4	t1000n2u2.txt	126	2.9784	139	2.3197
Execução nº 5	t1000n2u2.txt	144	2.7449	124	2.3702
Execução nº 6	t1000n2u2.txt	136	2.1065	128	2.3679
Execução nº 7	t1000n2u2.txt	135	3.6280	125	2.3851
Execução nº 8	t1000n2u2.txt	136	1.7948	138	2.3683
Execução nº 9	t1000n2u2.txt	140	1.9002	129	2.3685
Execução nº 10	t1000n2u2.txt	142	3.3406	127	2.3685
Execução nº 11	t1000n2u2.txt	146	2.8372	134	2.4132
Execução nº 12	t1000n2u2.txt	122	2.5385	132	2.3531
Execução nº 13	t1000n2u2.txt	130	4.3872	132	2.3371
Execução nº 14	t1000n2u2.txt	128	1.5977	133	2.3505
Execução nº 15	t1000n2u2.txt	100	1.6848	134	2.3343
Execução nº 16	t1000n2u2.txt	134	2.2701	132	2.4044
Execução nº 17	t1000n2u2.txt	107	2.0260	138	2.3662
Execução nº 18	t1000n2u2.txt	130	2.2474	144	2.3195
Execução nº 19	t1000n2u2.txt	128	2.7642	129	2.3042
Execução nº 20	t1000n2u2.txt	127	2.2576	121	2.4004
Execução nº 21	t1000n2u2.txt	128	1.8593	130	2.3364
Execução nº 22	t1000n2u2.txt	150	2.6176	131	2.3395
Execução nº 23	t1000n2u2.txt	128	3.0475	126	2.3528
Execução nº 24	t1000n2u2.txt	124	1.5622	136	2.3702
Execução nº 25	t1000n2u2.txt	139	2.8089	130	2.3606
Execução nº 26	t1000n2u2.txt	126	2.4012	132	2.3523
Execução nº 27	t1000n2u2.txt	149	3.5647	122	2.3871
Execução nº 28	t1000n2u2.txt	111	1.3365	136	2.3827
Execução nº 29	t1000n2u2.txt	122	4.6579	121	2.4170
Execução nº 30	t1000n2u2.txt	131	3.3905	135	2.3511

Fonte: Elaborada pelo autor (2026).

Tabela 15 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n2u4.txt	432	13.4801	419	6.4043
Execução n° 2	t1000n2u4.txt	446	27.1061	428	6.6049
Execução n° 3	t1000n2u4.txt	412	10.3651	410	6.8885
Execução n° 4	t1000n2u4.txt	460	23.4697	431	6.6818
Execução n° 5	t1000n2u4.txt	427	22.1626	436	6.5289
Execução n° 6	t1000n2u4.txt	493	44.0988	458	6.3994
Execução n° 7	t1000n2u4.txt	461	44.1473	417	7.0949
Execução n° 8	t1000n2u4.txt	467	22.9824	447	6.3835
Execução n° 9	t1000n2u4.txt	459	22.9123	437	6.7479
Execução n° 10	t1000n2u4.txt	459	44.0994	419	7.0950
Execução n° 11	t1000n2u4.txt	488	21.4120	445	6.3236
Execução n° 12	t1000n2u4.txt	463	27.3075	420	6.6068
Execução n° 13	t1000n2u4.txt	466	26.7446	420	6.8768
Execução n° 14	t1000n2u4.txt	458	17.8871	449	6.7034
Execução n° 15	t1000n2u4.txt	428	43.2684	424	6.4959
Execução n° 16	t1000n2u4.txt	482	37.4676	446	6.5254
Execução n° 17	t1000n2u4.txt	413	25.6963	435	6.7169
Execução n° 18	t1000n2u4.txt	453	34.2252	443	6.4787
Execução n° 19	t1000n2u4.txt	460	42.6388	412	6.6548
Execução n° 20	t1000n2u4.txt	449	23.0783	430	6.4941
Execução n° 21	t1000n2u4.txt	439	32.7794	441	6.8120
Execução n° 22	t1000n2u4.txt	462	19.1709	436	6.7146
Execução n° 23	t1000n2u4.txt	454	32.8248	440	6.4182
Execução n° 24	t1000n2u4.txt	447	21.0966	433	6.3287
Execução n° 25	t1000n2u4.txt	459	22.4040	431	6.4935
Execução n° 26	t1000n2u4.txt	479	36.8858	421	6.4638
Execução n° 27	t1000n2u4.txt	448	18.9612	443	6.6389
Execução n° 28	t1000n2u4.txt	491	33.3429	455	6.8261
Execução n° 29	t1000n2u4.txt	442	41.0903	425	6.9935
Execução n° 30	t1000n2u4.txt	457	26.0565	420	6.4343

Fonte: Elaborada pelo autor (2026).

Tabela 16 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n2u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n2u6.txt	752	98.4047	683	12.7378
Execução n° 2	t1000n2u6.txt	794	110.4037	725	12.3526
Execução n° 3	t1000n2u6.txt	761	99.0793	721	12.4775
Execução n° 4	t1000n2u6.txt	747	44.9037	686	13.2837
Execução n° 5	t1000n2u6.txt	774	106.8899	698	12.8557
Execução n° 6	t1000n2u6.txt	788	132.7865	728	12.7294
Execução n° 7	t1000n2u6.txt	752	119.5347	703	12.8962
Execução n° 8	t1000n2u6.txt	743	82.9019	664	12.4433
Execução n° 9	t1000n2u6.txt	783	50.2885	735	12.5708
Execução n° 10	t1000n2u6.txt	749	83.6687	698	13.4215
Execução n° 11	t1000n2u6.txt	768	75.2348	693	13.3081
Execução n° 12	t1000n2u6.txt	777	101.3361	698	13.1028
Execução n° 13	t1000n2u6.txt	769	78.1981	695	13.2600
Execução n° 14	t1000n2u6.txt	758	59.0526	713	12.9307
Execução n° 15	t1000n2u6.txt	769	140.5284	733	12.7759
Execução n° 16	t1000n2u6.txt	773	113.6524	719	12.9492
Execução n° 17	t1000n2u6.txt	750	120.2091	714	13.2705
Execução n° 18	t1000n2u6.txt	699	79.4017	705	13.1521
Execução n° 19	t1000n2u6.txt	768	81.3181	699	12.1971
Execução n° 20	t1000n2u6.txt	773	117.5128	717	13.2461
Execução n° 21	t1000n2u6.txt	717	104.3428	726	12.0568
Execução n° 22	t1000n2u6.txt	779	106.5794	694	12.3206
Execução n° 23	t1000n2u6.txt	746	72.5313	706	13.3812
Execução n° 24	t1000n2u6.txt	792	70.3249	744	12.9344
Execução n° 25	t1000n2u6.txt	773	97.1788	708	12.4330
Execução n° 26	t1000n2u6.txt	772	100.5550	710	12.8341
Execução n° 27	t1000n2u6.txt	784	95.4799	721	13.0398
Execução n° 28	t1000n2u6.txt	769	118.2191	722	13.0737
Execução n° 29	t1000n2u6.txt	714	153.5077	693	12.4744
Execução n° 30	t1000n2u6.txt	776	97.1826	698	12.5405

Fonte: Elaborada pelo autor (2026).

Tabela 17 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n3u2.txt	145	3.2563	151	2.3799
Execução n° 2	t1000n3u2.txt	145	2.5223	142	2.3090
Execução n° 3	t1000n3u2.txt	136	1.4905	136	2.2921
Execução n° 4	t1000n3u2.txt	148	2.9453	144	2.2915
Execução n° 5	t1000n3u2.txt	135	1.6000	141	2.3812
Execução n° 6	t1000n3u2.txt	140	2.5847	136	2.3229
Execução n° 7	t1000n3u2.txt	121	1.8635	133	2.3207
Execução n° 8	t1000n3u2.txt	150	2.5439	148	2.3212
Execução n° 9	t1000n3u2.txt	137	2.2100	137	2.3549
Execução n° 10	t1000n3u2.txt	148	2.9322	138	2.3040
Execução n° 11	t1000n3u2.txt	135	2.3500	135	2.3224
Execução n° 12	t1000n3u2.txt	150	4.4827	139	2.3683
Execução n° 13	t1000n3u2.txt	135	3.0438	137	2.3078
Execução n° 14	t1000n3u2.txt	139	2.5272	148	2.3181
Execução n° 15	t1000n3u2.txt	122	3.0856	142	2.3403
Execução n° 16	t1000n3u2.txt	129	4.0956	145	2.4026
Execução n° 17	t1000n3u2.txt	155	2.1149	136	2.3405
Execução n° 18	t1000n3u2.txt	140	3.7628	135	2.3226
Execução n° 19	t1000n3u2.txt	130	1.7402	140	2.3845
Execução n° 20	t1000n3u2.txt	141	1.2083	149	2.3069
Execução n° 21	t1000n3u2.txt	132	5.6123	148	2.3218
Execução n° 22	t1000n3u2.txt	143	2.0582	143	2.4008
Execução n° 23	t1000n3u2.txt	129	1.1126	151	2.3374
Execução n° 24	t1000n3u2.txt	152	1.9495	143	2.3295
Execução n° 25	t1000n3u2.txt	141	2.3200	136	2.2925
Execução n° 26	t1000n3u2.txt	136	2.6466	140	2.3132
Execução n° 27	t1000n3u2.txt	136	1.8181	134	2.3691
Execução n° 28	t1000n3u2.txt	129	1.9119	144	2.3378
Execução n° 29	t1000n3u2.txt	138	3.4827	140	2.3233
Execução n° 30	t1000n3u2.txt	116	2.0664	139	2.3025

Fonte: Elaborada pelo autor (2026).

Tabela 18 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n3u4.txt	402	17.2464	376	6.9224
Execução nº 2	t1000n3u4.txt	418	20.0934	369	7.1843
Execução nº 3	t1000n3u4.txt	416	28.0600	379	6.9836
Execução nº 4	t1000n3u4.txt	438	17.9195	413	6.8848
Execução nº 5	t1000n3u4.txt	409	18.4175	394	6.8421
Execução nº 6	t1000n3u4.txt	428	29.9080	393	6.8270
Execução nº 7	t1000n3u4.txt	379	36.6526	395	6.9828
Execução nº 8	t1000n3u4.txt	426	33.7044	398	6.9975
Execução nº 9	t1000n3u4.txt	387	18.7197	391	7.0952
Execução nº 10	t1000n3u4.txt	365	27.5556	374	7.1252
Execução nº 11	t1000n3u4.txt	422	16.1861	403	6.9831
Execução nº 12	t1000n3u4.txt	417	36.5887	406	6.7176
Execução nº 13	t1000n3u4.txt	422	24.5307	389	7.0951
Execução nº 14	t1000n3u4.txt	410	23.7995	364	7.0299
Execução nº 15	t1000n3u4.txt	413	40.6054	392	6.9695
Execução nº 16	t1000n3u4.txt	425	27.2132	382	6.9053
Execução nº 17	t1000n3u4.txt	376	32.0674	384	6.9816
Execução nº 18	t1000n3u4.txt	349	17.7283	407	6.8549
Execução nº 19	t1000n3u4.txt	416	29.3162	369	6.8739
Execução nº 20	t1000n3u4.txt	409	28.7643	391	6.9723
Execução nº 21	t1000n3u4.txt	389	20.0375	387	6.7939
Execução nº 22	t1000n3u4.txt	410	22.5932	396	6.9333
Execução nº 23	t1000n3u4.txt	368	31.0895	402	6.8264
Execução nº 24	t1000n3u4.txt	419	26.9676	380	6.8129
Execução nº 25	t1000n3u4.txt	379	21.3171	381	7.1215
Execução nº 26	t1000n3u4.txt	401	36.0666	365	7.0301
Execução nº 27	t1000n3u4.txt	424	21.6030	401	6.8125
Execução nº 28	t1000n3u4.txt	424	30.8050	403	7.0125
Execução nº 29	t1000n3u4.txt	364	34.4168	393	7.1248
Execução nº 30	t1000n3u4.txt	401	22.2281	381	6.9378

Fonte: Elaborada pelo autor (2026).

Tabela 19 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n3u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n3u6.txt	771	138.7388	703	12.8855
Execução nº 2	t1000n3u6.txt	770	79.8660	709	13.4770
Execução nº 3	t1000n3u6.txt	701	98.5776	718	13.2945
Execução nº 4	t1000n3u6.txt	686	35.5403	724	12.4574
Execução nº 5	t1000n3u6.txt	761	81.2049	694	13.3858
Execução nº 6	t1000n3u6.txt	772	117.4470	690	12.8978
Execução nº 7	t1000n3u6.txt	722	141.7962	685	12.6684
Execução nº 8	t1000n3u6.txt	775	98.7472	705	12.4415
Execução nº 9	t1000n3u6.txt	743	45.5728	708	11.9895
Execução nº 10	t1000n3u6.txt	764	107.8442	709	14.0586
Execução nº 11	t1000n3u6.txt	760	121.6542	703	13.3101
Execução nº 12	t1000n3u6.txt	761	104.3758	717	13.1552
Execução nº 13	t1000n3u6.txt	748	74.4481	722	12.3537
Execução nº 14	t1000n3u6.txt	778	60.2327	709	12.5556
Execução nº 15	t1000n3u6.txt	777	179.1931	729	12.8676
Execução nº 16	t1000n3u6.txt	790	151.0951	716	13.2304
Execução nº 17	t1000n3u6.txt	757	128.5307	717	13.2473
Execução nº 18	t1000n3u6.txt	727	84.1819	670	13.0545
Execução nº 19	t1000n3u6.txt	755	64.5621	697	12.9518
Execução nº 20	t1000n3u6.txt	718	126.8721	716	12.5994
Execução nº 21	t1000n3u6.txt	614	123.7682	729	11.9938
Execução nº 22	t1000n3u6.txt	770	73.0844	702	12.5848
Execução nº 23	t1000n3u6.txt	718	80.3373	698	12.7721
Execução nº 24	t1000n3u6.txt	776	54.1162	709	12.2273
Execução nº 25	t1000n3u6.txt	753	110.2332	700	12.6663
Execução nº 26	t1000n3u6.txt	757	81.8340	704	12.8212
Execução nº 27	t1000n3u6.txt	775	72.0645	689	11.9146
Execução nº 28	t1000n3u6.txt	735	113.8286	736	12.3054
Execução nº 29	t1000n3u6.txt	743	104.8933	692	13.4022
Execução nº 30	t1000n3u6.txt	740	81.2606	718	13.1194

Fonte: Elaborada pelo autor (2026).

Tabela 20 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n4u2.txt	171	3.6507	164	2.4142
Execução n° 2	t1000n4u2.txt	168	3.0006	164	2.2710
Execução n° 3	t1000n4u2.txt	165	1.4433	174	2.2572
Execução n° 4	t1000n4u2.txt	171	2.0046	160	2.2407
Execução n° 5	t1000n4u2.txt	177	1.7502	170	2.3393
Execução n° 6	t1000n4u2.txt	169	1.9356	175	2.3214
Execução n° 7	t1000n4u2.txt	117	1.9065	169	2.2451
Execução n° 8	t1000n4u2.txt	171	1.9240	170	2.3028
Execução n° 9	t1000n4u2.txt	168	3.7466	164	2.3240
Execução n° 10	t1000n4u2.txt	171	2.9170	168	2.2549
Execução n° 11	t1000n4u2.txt	169	1.9199	169	2.1672
Execução n° 12	t1000n4u2.txt	162	2.0981	165	2.2608
Execução n° 13	t1000n4u2.txt	165	4.2382	159	2.3042
Execução n° 14	t1000n4u2.txt	171	1.9141	171	2.3544
Execução n° 15	t1000n4u2.txt	143	8.0418	160	2.3076
Execução n° 16	t1000n4u2.txt	176	5.3795	170	2.2145
Execução n° 17	t1000n4u2.txt	154	2.2375	168	2.2409
Execução n° 18	t1000n4u2.txt	184	5.4047	167	2.2450
Execução n° 19	t1000n4u2.txt	165	4.6739	172	2.2153
Execução n° 20	t1000n4u2.txt	160	1.4671	165	2.2273
Execução n° 21	t1000n4u2.txt	138	2.9781	163	2.2776
Execução n° 22	t1000n4u2.txt	176	2.7758	175	2.3340
Execução n° 23	t1000n4u2.txt	174	1.7992	175	2.2929
Execução n° 24	t1000n4u2.txt	166	1.4029	178	2.2738
Execução n° 25	t1000n4u2.txt	175	1.3082	166	2.2759
Execução n° 26	t1000n4u2.txt	154	2.2086	169	2.2561
Execução n° 27	t1000n4u2.txt	175	3.4521	176	2.2914
Execução n° 28	t1000n4u2.txt	155	4.0815	179	2.2005
Execução n° 29	t1000n4u2.txt	172	6.5797	167	2.3862
Execução n° 30	t1000n4u2.txt	157	6.6019	173	3.0039

Fonte: Elaborada pelo autor (2026).

Tabela 21 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n4u4.txt	430	28.4590	432	6.6998
Execução n° 2	t1000n4u4.txt	422	16.0413	395	6.8244
Execução n° 3	t1000n4u4.txt	420	15.8731	397	6.5148
Execução n° 4	t1000n4u4.txt	443	22.9030	414	6.7483
Execução n° 5	t1000n4u4.txt	436	16.9801	402	7.0320
Execução n° 6	t1000n4u4.txt	448	39.1023	402	6.8549
Execução n° 7	t1000n4u4.txt	424	31.7169	426	7.0277
Execução n° 8	t1000n4u4.txt	461	29.2770	411	6.6408
Execução n° 9	t1000n4u4.txt	420	12.2232	415	6.8129
Execução n° 10	t1000n4u4.txt	434	23.0014	427	6.9539
Execução n° 11	t1000n4u4.txt	466	20.5760	402	7.1545
Execução n° 12	t1000n4u4.txt	450	40.5682	432	6.7737
Execução n° 13	t1000n4u4.txt	452	31.2878	433	6.6443
Execução n° 14	t1000n4u4.txt	402	9.1652	416	7.1413
Execução n° 15	t1000n4u4.txt	443	54.4518	412	6.7381
Execução n° 16	t1000n4u4.txt	435	31.8082	397	6.6235
Execução n° 17	t1000n4u4.txt	405	18.5314	403	6.5774
Execução n° 18	t1000n4u4.txt	429	27.0271	427	6.7454
Execução n° 19	t1000n4u4.txt	440	21.0902	398	6.9339
Execução n° 20	t1000n4u4.txt	419	16.0199	396	6.6825
Execução n° 21	t1000n4u4.txt	419	29.1447	416	6.6180
Execução n° 22	t1000n4u4.txt	446	14.4370	406	7.0414
Execução n° 23	t1000n4u4.txt	388	12.2046	410	6.8119
Execução n° 24	t1000n4u4.txt	438	21.2099	383	6.4639
Execução n° 25	t1000n4u4.txt	378	12.9306	418	6.5576
Execução n° 26	t1000n4u4.txt	456	37.8578	394	7.0119
Execução n° 27	t1000n4u4.txt	417	17.6353	394	6.6392
Execução n° 28	t1000n4u4.txt	459	36.2059	418	6.6188
Execução n° 29	t1000n4u4.txt	425	33.8024	409	6.9252
Execução n° 30	t1000n4u4.txt	438	25.3916	411	6.3247

Fonte: Elaborada pelo autor (2026).

Tabela 22 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n4u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n4u6.txt	813	135.3155	728	12.6145
Execução n° 2	t1000n4u6.txt	791	59.0548	743	12.0133
Execução n° 3	t1000n4u6.txt	789	110.3966	697	12.1314
Execução n° 4	t1000n4u6.txt	760	59.4822	728	12.4020
Execução n° 5	t1000n4u6.txt	789	115.8539	711	12.1022
Execução n° 6	t1000n4u6.txt	768	160.9398	725	13.0577
Execução n° 7	t1000n4u6.txt	806	96.7602	721	12.7295
Execução n° 8	t1000n4u6.txt	795	80.7435	734	13.1507
Execução n° 9	t1000n4u6.txt	781	67.8492	709	13.2801
Execução n° 10	t1000n4u6.txt	763	122.0828	745	11.9906
Execução n° 11	t1000n4u6.txt	783	57.4035	727	13.1179
Execução n° 12	t1000n4u6.txt	785	97.8951	712	12.1411
Execução n° 13	t1000n4u6.txt	786	104.9726	736	12.4178
Execução n° 14	t1000n4u6.txt	768	132.2638	746	11.8066
Execução n° 15	t1000n4u6.txt	776	138.6948	753	12.6970
Execução n° 16	t1000n4u6.txt	776	99.5338	753	13.0850
Execução n° 17	t1000n4u6.txt	759	82.4654	745	12.2389
Execução n° 18	t1000n4u6.txt	767	97.7765	704	12.0226
Execução n° 19	t1000n4u6.txt	721	86.9923	719	12.3392
Execução n° 20	t1000n4u6.txt	713	59.5393	752	12.3133
Execução n° 21	t1000n4u6.txt	778	105.0133	746	13.4922
Execução n° 22	t1000n4u6.txt	787	113.9823	738	13.4829
Execução n° 23	t1000n4u6.txt	705	65.7316	737	12.1497
Execução n° 24	t1000n4u6.txt	752	75.5564	692	12.8441
Execução n° 25	t1000n4u6.txt	750	112.1037	737	13.3232
Execução n° 26	t1000n4u6.txt	764	101.7011	725	13.4986
Execução n° 27	t1000n4u6.txt	798	66.2377	750	12.4460
Execução n° 28	t1000n4u6.txt	742	138.2848	724	12.3245
Execução n° 29	t1000n4u6.txt	769	121.5665	680	11.9426
Execução n° 30	t1000n4u6.txt	777	91.7488	730	12.3077

Fonte: Elaborada pelo autor (2026).

Tabela 23 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n5u2.txt	161	2.8569	155	2.3012
Execução nº 2	t1000n5u2.txt	149	1.7880	162	2.1952
Execução nº 3	t1000n5u2.txt	169	1.9442	165	2.3599
Execução nº 4	t1000n5u2.txt	169	1.7866	167	2.2590
Execução nº 5	t1000n5u2.txt	175	1.9657	171	2.3062
Execução nº 6	t1000n5u2.txt	163	3.0477	170	2.3303
Execução nº 7	t1000n5u2.txt	142	2.1162	169	2.2763
Execução nº 8	t1000n5u2.txt	171	1.8124	172	2.2910
Execução nº 9	t1000n5u2.txt	156	2.1957	159	2.2573
Execução nº 10	t1000n5u2.txt	162	3.6359	160	2.2766
Execução nº 11	t1000n5u2.txt	160	1.4182	156	2.2401
Execução nº 12	t1000n5u2.txt	153	3.2718	164	2.2913
Execução nº 13	t1000n5u2.txt	149	3.1653	164	2.3037
Execução nº 14	t1000n5u2.txt	176	3.1678	160	2.3228
Execução nº 15	t1000n5u2.txt	151	7.2120	164	2.3062
Execução nº 16	t1000n5u2.txt	166	4.0316	166	2.2597
Execução nº 17	t1000n5u2.txt	162	2.0085	171	2.2889
Execução nº 18	t1000n5u2.txt	157	4.1489	164	2.3146
Execução nº 19	t1000n5u2.txt	161	2.5217	145	2.4081
Execução nº 20	t1000n5u2.txt	154	2.3811	159	2.2508
Execução nº 21	t1000n5u2.txt	137	3.7057	159	2.3024
Execução nº 22	t1000n5u2.txt	160	2.0523	165	2.3843
Execução nº 23	t1000n5u2.txt	157	1.8351	161	2.2729
Execução nº 24	t1000n5u2.txt	177	3.3713	163	2.2347
Execução nº 25	t1000n5u2.txt	166	1.6784	164	2.2634
Execução nº 26	t1000n5u2.txt	156	4.5918	161	2.2454
Execução nº 27	t1000n5u2.txt	159	3.2144	149	2.2763
Execução nº 28	t1000n5u2.txt	160	6.7391	156	2.2916
Execução nº 29	t1000n5u2.txt	175	4.5031	161	2.2584
Execução nº 30	t1000n5u2.txt	165	5.7414	160	2.2281

Fonte: Elaborada pelo autor (2026).

Tabela 24 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n5u4.txt	398	35.6973	387	7.0624
Execução nº 2	t1000n5u4.txt	428	41.2758	381	7.0429
Execução nº 3	t1000n5u4.txt	377	13.8656	382	7.0624
Execução nº 4	t1000n5u4.txt	410	14.3767	341	7.0731
Execução nº 5	t1000n5u4.txt	396	27.4850	362	7.1050
Execução nº 6	t1000n5u4.txt	405	18.5003	397	6.8737
Execução nº 7	t1000n5u4.txt	390	44.9448	362	6.9370
Execução nº 8	t1000n5u4.txt	397	21.2511	384	6.8907
Execução nº 9	t1000n5u4.txt	416	14.6527	356	6.8354
Execução nº 10	t1000n5u4.txt	412	19.0971	364	7.3015
Execução nº 11	t1000n5u4.txt	408	26.8555	371	7.0326
Execução nº 12	t1000n5u4.txt	384	18.1322	378	6.8069
Execução nº 13	t1000n5u4.txt	397	20.1916	374	7.2766
Execução nº 14	t1000n5u4.txt	419	15.1901	369	7.0108
Execução nº 15	t1000n5u4.txt	377	23.1685	368	7.2368
Execução nº 16	t1000n5u4.txt	366	25.5787	376	6.9202
Execução nº 17	t1000n5u4.txt	360	25.1491	381	6.9868
Execução nº 18	t1000n5u4.txt	404	30.1367	362	6.7783
Execução nº 19	t1000n5u4.txt	368	16.6592	389	6.9181
Execução nº 20	t1000n5u4.txt	348	8.7272	355	6.7788
Execução nº 21	t1000n5u4.txt	344	35.6903	365	6.8598
Execução nº 22	t1000n5u4.txt	412	23.5926	366	7.1552
Execução nº 23	t1000n5u4.txt	422	25.6484	387	7.1563
Execução nº 24	t1000n5u4.txt	428	19.1122	392	7.1439
Execução nº 25	t1000n5u4.txt	359	15.1390	367	7.5316
Execução nº 26	t1000n5u4.txt	390	25.8648	404	7.0614
Execução nº 27	t1000n5u4.txt	338	20.6724	369	7.0609
Execução nº 28	t1000n5u4.txt	361	18.5781	398	6.7148
Execução nº 29	t1000n5u4.txt	371	28.5633	365	6.9056
Execução nº 30	t1000n5u4.txt	405	21.0296	390	7.1534

Fonte: Elaborada pelo autor (2026).

Tabela 25 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n5u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n5u6.txt	757	112.6471	748	13.0193
Execução nº 2	t1000n5u6.txt	779	103.0432	724	13.7166
Execução nº 3	t1000n5u6.txt	749	93.6709	745	13.0731
Execução nº 4	t1000n5u6.txt	725	44.9755	701	13.5102
Execução nº 5	t1000n5u6.txt	776	89.3729	741	12.6523
Execução nº 6	t1000n5u6.txt	775	91.9169	727	12.9966
Execução nº 7	t1000n5u6.txt	741	106.1455	702	13.0898
Execução nº 8	t1000n5u6.txt	817	82.8593	751	12.6701
Execução nº 9	t1000n5u6.txt	749	37.5932	702	12.5301
Execução nº 10	t1000n5u6.txt	770	84.6685	730	13.2621
Execução nº 11	t1000n5u6.txt	770	83.6718	722	13.1349
Execução nº 12	t1000n5u6.txt	807	99.6790	721	12.8903
Execução nº 13	t1000n5u6.txt	798	86.7579	719	13.0300
Execução nº 14	t1000n5u6.txt	750	69.7759	723	12.6018
Execução nº 15	t1000n5u6.txt	768	117.4950	707	13.1969
Execução nº 16	t1000n5u6.txt	739	111.2142	734	12.5407
Execução nº 17	t1000n5u6.txt	761	96.6617	705	12.9743
Execução nº 18	t1000n5u6.txt	726	85.9565	700	12.6182
Execução nº 19	t1000n5u6.txt	772	106.6531	680	13.4199
Execução nº 20	t1000n5u6.txt	715	86.7713	710	13.1837
Execução nº 21	t1000n5u6.txt	706	135.3714	704	12.0355
Execução nº 22	t1000n5u6.txt	800	69.4946	721	12.6508
Execução nº 23	t1000n5u6.txt	739	89.9861	657	13.4128
Execução nº 24	t1000n5u6.txt	776	80.1883	709	13.0291
Execução nº 25	t1000n5u6.txt	757	112.0604	716	12.7919
Execução nº 26	t1000n5u6.txt	769	84.2444	742	12.5717
Execução nº 27	t1000n5u6.txt	787	93.7316	667	12.6352
Execução nº 28	t1000n5u6.txt	790	104.0135	701	13.0238
Execução nº 29	t1000n5u6.txt	763	116.6079	708	12.5044
Execução nº 30	t1000n5u6.txt	772	86.0669	743	13.3603

Fonte: Elaborada pelo autor (2026).

Tabela 26 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n6u2.txt	148	2.1115	141	2.4269
Execução nº 2	t1000n6u2.txt	147	1.9411	140	2.3625
Execução nº 3	t1000n6u2.txt	132	1.4904	141	2.3808
Execução nº 4	t1000n6u2.txt	129	1.7246	138	2.4343
Execução nº 5	t1000n6u2.txt	147	3.2288	156	2.3208
Execução nº 6	t1000n6u2.txt	126	1.0997	141	2.3637
Execução nº 7	t1000n6u2.txt	136	3.3346	132	2.3382
Execução nº 8	t1000n6u2.txt	141	2.8389	152	2.3639
Execução nº 9	t1000n6u2.txt	152	1.9905	144	2.3572
Execução nº 10	t1000n6u2.txt	161	3.1662	148	2.3352
Execução nº 11	t1000n6u2.txt	158	1.7241	149	2.3397
Execução nº 12	t1000n6u2.txt	136	1.9444	133	2.3750
Execução nº 13	t1000n6u2.txt	130	2.5211	148	2.3356
Execução nº 14	t1000n6u2.txt	121	1.6785	157	2.3236
Execução nº 15	t1000n6u2.txt	105	1.7436	142	2.3830
Execução nº 16	t1000n6u2.txt	139	2.2725	152	2.3223
Execução nº 17	t1000n6u2.txt	118	1.5038	124	2.3017
Execução nº 18	t1000n6u2.txt	123	2.5706	139	2.3577
Execução nº 19	t1000n6u2.txt	129	2.5662	138	2.3546
Execução nº 20	t1000n6u2.txt	161	3.1181	140	2.4088
Execução nº 21	t1000n6u2.txt	139	4.2178	138	2.3536
Execução nº 22	t1000n6u2.txt	162	4.0111	148	2.3234
Execução nº 23	t1000n6u2.txt	121	2.3678	138	2.3614
Execução nº 24	t1000n6u2.txt	126	1.8521	134	2.3398
Execução nº 25	t1000n6u2.txt	150	2.8527	135	2.3520
Execução nº 26	t1000n6u2.txt	157	1.8457	138	2.4160
Execução nº 27	t1000n6u2.txt	157	3.1609	130	2.3071
Execução nº 28	t1000n6u2.txt	154	4.1058	159	2.3530
Execução nº 29	t1000n6u2.txt	112	2.9751	150	2.3399
Execução nº 30	t1000n6u2.txt	130	1.8496	154	2.3506

Fonte: Elaborada pelo autor (2026).

Tabela 27 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n6u4.txt	451	35.5828	443	6.7477
Execução n° 2	t1000n6u4.txt	459	43.0176	447	6.8422
Execução n° 3	t1000n6u4.txt	443	18.0440	443	6.4784
Execução n° 4	t1000n6u4.txt	454	13.9970	410	6.8302
Execução n° 5	t1000n6u4.txt	406	14.0917	404	6.9992
Execução n° 6	t1000n6u4.txt	466	20.8810	426	6.9191
Execução n° 7	t1000n6u4.txt	429	37.2977	406	6.5885
Execução n° 8	t1000n6u4.txt	459	20.5584	403	6.7025
Execução n° 9	t1000n6u4.txt	461	15.2439	402	6.6195
Execução n° 10	t1000n6u4.txt	463	21.8400	420	6.6115
Execução n° 11	t1000n6u4.txt	462	24.3107	408	6.8529
Execução n° 12	t1000n6u4.txt	447	23.7793	404	6.7358
Execução n° 13	t1000n6u4.txt	467	41.0557	403	6.6407
Execução n° 14	t1000n6u4.txt	444	17.6597	442	6.5737
Execução n° 15	t1000n6u4.txt	412	34.7204	423	6.7291
Execução n° 16	t1000n6u4.txt	462	34.7544	406	6.7922
Execução n° 17	t1000n6u4.txt	456	49.3947	401	6.5480
Execução n° 18	t1000n6u4.txt	418	28.5674	444	6.8727
Execução n° 19	t1000n6u4.txt	444	29.9002	427	7.1582
Execução n° 20	t1000n6u4.txt	458	17.1654	447	6.9887
Execução n° 21	t1000n6u4.txt	407	26.6942	410	6.6210
Execução n° 22	t1000n6u4.txt	460	42.5559	426	7.3294
Execução n° 23	t1000n6u4.txt	454	23.6376	434	6.6265
Execução n° 24	t1000n6u4.txt	469	24.2541	413	6.5801
Execução n° 25	t1000n6u4.txt	459	25.3696	383	6.9696
Execução n° 26	t1000n6u4.txt	437	29.8038	433	6.8121
Execução n° 27	t1000n6u4.txt	460	37.1996	381	6.9695
Execução n° 28	t1000n6u4.txt	464	21.5193	396	6.8267
Execução n° 29	t1000n6u4.txt	432	48.7090	417	6.9049
Execução n° 30	t1000n6u4.txt	411	36.9176	440	6.6056

Fonte: Elaborada pelo autor (2026).

Tabela 28 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n6u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n6u6.txt	771	100.4123	732	13.5494
Execução nº 2	t1000n6u6.txt	774	73.4131	715	12.5366
Execução nº 3	t1000n6u6.txt	778	152.0519	760	12.2971
Execução nº 4	t1000n6u6.txt	774	55.5796	705	12.8538
Execução nº 5	t1000n6u6.txt	775	106.3166	713	12.5101
Execução nº 6	t1000n6u6.txt	770	122.4888	706	12.5268
Execução nº 7	t1000n6u6.txt	781	106.2022	728	12.4767
Execução nº 8	t1000n6u6.txt	766	68.3704	747	12.6826
Execução nº 9	t1000n6u6.txt	786	76.3330	676	12.4810
Execução nº 10	t1000n6u6.txt	721	82.3513	677	12.9319
Execução nº 11	t1000n6u6.txt	783	62.9948	711	13.0725
Execução nº 12	t1000n6u6.txt	769	94.1244	718	13.2430
Execução nº 13	t1000n6u6.txt	760	77.7663	683	12.7062
Execução nº 14	t1000n6u6.txt	785	93.8421	735	12.1656
Execução nº 15	t1000n6u6.txt	795	141.5262	734	13.4202
Execução nº 16	t1000n6u6.txt	773	84.8018	723	11.8202
Execução nº 17	t1000n6u6.txt	684	80.5922	744	13.6379
Execução nº 18	t1000n6u6.txt	750	128.9815	714	12.7744
Execução nº 19	t1000n6u6.txt	726	72.4898	716	13.4048
Execução nº 20	t1000n6u6.txt	775	81.1450	754	12.1808
Execução nº 21	t1000n6u6.txt	711	101.6377	730	13.3569
Execução nº 22	t1000n6u6.txt	761	85.1846	735	12.2723
Execução nº 23	t1000n6u6.txt	772	93.3152	730	13.0717
Execução nº 24	t1000n6u6.txt	774	58.5164	727	12.7587
Execução nº 25	t1000n6u6.txt	772	82.3403	691	12.9254
Execução nº 26	t1000n6u6.txt	765	91.7482	698	12.5698
Execução nº 27	t1000n6u6.txt	794	56.8972	742	12.8717
Execução nº 28	t1000n6u6.txt	779	91.4738	759	12.3993
Execução nº 29	t1000n6u6.txt	774	147.5495	700	12.3810
Execução nº 30	t1000n6u6.txt	783	170.0586	712	12.8534

Fonte: Elaborada pelo autor (2026).

Tabela 29 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n7u2.txt	137	3.4106	142	2.3255
Execução nº 2	t1000n7u2.txt	131	1.8315	135	2.2897
Execução nº 3	t1000n7u2.txt	153	2.0861	151	2.2619
Execução nº 4	t1000n7u2.txt	154	1.8743	157	2.3161
Execução nº 5	t1000n7u2.txt	153	1.5252	150	2.3552
Execução nº 6	t1000n7u2.txt	160	1.8114	147	2.3946
Execução nº 7	t1000n7u2.txt	124	3.0521	146	2.3234
Execução nº 8	t1000n7u2.txt	154	2.0689	149	2.3224
Execução nº 9	t1000n7u2.txt	156	3.5713	146	2.3205
Execução nº 10	t1000n7u2.txt	151	2.2551	148	2.3563
Execução nº 11	t1000n7u2.txt	150	4.0141	144	2.2967
Execução nº 12	t1000n7u2.txt	148	1.7857	140	2.3986
Execução nº 13	t1000n7u2.txt	146	2.6786	147	2.3229
Execução nº 14	t1000n7u2.txt	158	2.0228	152	2.3819
Execução nº 15	t1000n7u2.txt	140	4.6471	150	2.3072
Execução nº 16	t1000n7u2.txt	141	3.1350	145	2.3053
Execução nº 17	t1000n7u2.txt	148	1.8501	158	2.2757
Execução nº 18	t1000n7u2.txt	161	5.9865	142	2.3980
Execução nº 19	t1000n7u2.txt	155	4.1489	143	2.3855
Execução nº 20	t1000n7u2.txt	154	1.8188	145	2.3402
Execução nº 21	t1000n7u2.txt	115	1.8797	142	2.3245
Execução nº 22	t1000n7u2.txt	151	2.2827	151	2.3030
Execução nº 23	t1000n7u2.txt	149	2.4916	151	2.3086
Execução nº 24	t1000n7u2.txt	155	1.2199	152	2.2752
Execução nº 25	t1000n7u2.txt	153	1.4426	151	2.4051
Execução nº 26	t1000n7u2.txt	140	3.8714	151	2.4400
Execução nº 27	t1000n7u2.txt	125	1.1124	151	2.3334
Execução nº 28	t1000n7u2.txt	150	2.9267	151	2.4488
Execução nº 29	t1000n7u2.txt	147	3.4826	151	2.3503
Execução nº 30	t1000n7u2.txt	147	4.5921	148	2.3240

Fonte: Elaborada pelo autor (2026).

Tabela 30 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução n° 1	t1000n7u4.txt	461	45.6959	438	6.7133
Execução n° 2	t1000n7u4.txt	453	37.3922	427	6.4579
Execução n° 3	t1000n7u4.txt	426	19.8925	412	6.4963
Execução n° 4	t1000n7u4.txt	430	8.1985	439	6.5699
Execução n° 5	t1000n7u4.txt	449	32.0581	393	7.0467
Execução n° 6	t1000n7u4.txt	459	17.0713	425	6.5891
Execução n° 7	t1000n7u4.txt	407	48.5504	403	6.9050
Execução n° 8	t1000n7u4.txt	429	19.8588	411	6.3398
Execução n° 9	t1000n7u4.txt	456	19.0259	427	6.4985
Execução n° 10	t1000n7u4.txt	455	24.0028	416	7.1099
Execução n° 11	t1000n7u4.txt	441	27.1841	398	7.2523
Execução n° 12	t1000n7u4.txt	433	33.0902	428	6.8593
Execução n° 13	t1000n7u4.txt	470	35.2866	425	6.7041
Execução n° 14	t1000n7u4.txt	448	7.9635	402	6.7684
Execução n° 15	t1000n7u4.txt	431	46.5023	427	6.6082
Execução n° 16	t1000n7u4.txt	477	50.1387	416	6.7598
Execução n° 17	t1000n7u4.txt	427	26.3670	430	6.7630
Execução n° 18	t1000n7u4.txt	455	44.7871	436	6.6663
Execução n° 19	t1000n7u4.txt	465	23.0264	422	6.5748
Execução n° 20	t1000n7u4.txt	417	22.8211	438	7.0143
Execução n° 21	t1000n7u4.txt	365	32.4509	436	6.5259
Execução n° 22	t1000n7u4.txt	456	20.6790	418	6.6699
Execução n° 23	t1000n7u4.txt	429	18.8738	434	7.1428
Execução n° 24	t1000n7u4.txt	478	26.8248	411	6.5618
Execução n° 25	t1000n7u4.txt	454	26.0063	408	6.7492
Execução n° 26	t1000n7u4.txt	447	29.8096	410	6.4045
Execução n° 27	t1000n7u4.txt	423	30.6948	424	6.8510
Execução n° 28	t1000n7u4.txt	435	32.1011	393	6.6391
Execução n° 29	t1000n7u4.txt	438	40.7440	394	6.7463
Execução n° 30	t1000n7u4.txt	461	29.3946	396	6.3431

Fonte: Elaborada pelo autor (2026).

Tabela 31 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n7u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n7u6.txt	758	101.1906	710	13.5820
Execução nº 2	t1000n7u6.txt	785	100.3266	604	13.0380
Execução nº 3	t1000n7u6.txt	762	97.9898	673	12.7929
Execução nº 4	t1000n7u6.txt	732	44.2060	703	13.9849
Execução nº 5	t1000n7u6.txt	737	63.2043	695	12.9506
Execução nº 6	t1000n7u6.txt	757	167.0129	691	13.4320
Execução nº 7	t1000n7u6.txt	744	114.5456	712	13.4027
Execução nº 8	t1000n7u6.txt	768	104.9822	679	12.6401
Execução nº 9	t1000n7u6.txt	731	59.0128	696	12.9026
Execução nº 10	t1000n7u6.txt	773	145.2079	685	13.2939
Execução nº 11	t1000n7u6.txt	775	94.2600	700	12.7712
Execução nº 12	t1000n7u6.txt	746	96.9689	694	13.2620
Execução nº 13	t1000n7u6.txt	753	85.2305	635	13.5622
Execução nº 14	t1000n7u6.txt	741	62.0627	697	12.9628
Execução nº 15	t1000n7u6.txt	757	212.7988	658	13.1651
Execução nº 16	t1000n7u6.txt	752	91.8041	678	13.5437
Execução nº 17	t1000n7u6.txt	739	100.7198	713	12.7612
Execução nº 18	t1000n7u6.txt	730	95.3904	695	13.1212
Execução nº 19	t1000n7u6.txt	752	142.9036	713	13.6686
Execução nº 20	t1000n7u6.txt	730	83.4744	660	13.4480
Execução nº 21	t1000n7u6.txt	706	126.5077	676	12.6800
Execução nº 22	t1000n7u6.txt	786	108.2770	674	13.6059
Execução nº 23	t1000n7u6.txt	729	89.6251	703	13.6084
Execução nº 24	t1000n7u6.txt	741	91.8329	686	13.5523
Execução nº 25	t1000n7u6.txt	724	94.5808	699	13.1380
Execução nº 26	t1000n7u6.txt	771	79.8544	660	13.1822
Execução nº 27	t1000n7u6.txt	769	80.8513	714	12.5277
Execução nº 28	t1000n7u6.txt	738	105.7879	705	12.5020
Execução nº 29	t1000n7u6.txt	736	112.3374	689	13.3534
Execução nº 30	t1000n7u6.txt	763	99.5790	705	13.2306

Fonte: Elaborada pelo autor (2026).

Tabela 32 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n8u2.txt	142	3.9366	143	2.4100
Execução nº 2	t1000n8u2.txt	148	2.2416	145	2.3341
Execução nº 3	t1000n8u2.txt	135	1.1270	146	2.3990
Execução nº 4	t1000n8u2.txt	135	1.9260	136	2.3525
Execução nº 5	t1000n8u2.txt	134	2.3079	144	2.3703
Execução nº 6	t1000n8u2.txt	134	3.1822	146	2.4378
Execução nº 7	t1000n8u2.txt	127	2.7288	130	2.3552
Execução nº 8	t1000n8u2.txt	146	2.1756	125	2.4006
Execução nº 9	t1000n8u2.txt	152	2.9671	137	2.4014
Execução nº 10	t1000n8u2.txt	157	3.0720	135	2.3082
Execução nº 11	t1000n8u2.txt	135	2.7596	135	2.3350
Execução nº 12	t1000n8u2.txt	116	1.9937	131	2.3653
Execução nº 13	t1000n8u2.txt	117	2.7290	133	2.3668
Execução nº 14	t1000n8u2.txt	145	2.8744	139	2.3533
Execução nº 15	t1000n8u2.txt	125	3.2451	121	2.3220
Execução nº 16	t1000n8u2.txt	136	6.9127	146	2.3183
Execução nº 17	t1000n8u2.txt	137	1.2185	140	2.3730
Execução nº 18	t1000n8u2.txt	123	2.1962	133	2.3837
Execução nº 19	t1000n8u2.txt	118	1.4716	129	2.3236
Execução nº 20	t1000n8u2.txt	123	1.3185	138	2.3075
Execução nº 21	t1000n8u2.txt	131	4.1970	119	2.3461
Execução nº 22	t1000n8u2.txt	145	1.9574	140	2.3244
Execução nº 23	t1000n8u2.txt	145	1.7389	138	2.3046
Execução nº 24	t1000n8u2.txt	148	1.6360	142	2.2925
Execução nº 25	t1000n8u2.txt	127	1.8961	138	2.3355
Execução nº 26	t1000n8u2.txt	137	3.7760	123	2.3235
Execução nº 27	t1000n8u2.txt	140	3.1765	138	2.3355
Execução nº 28	t1000n8u2.txt	112	1.1500	132	2.3395
Execução nº 29	t1000n8u2.txt	133	2.1168	137	2.3684
Execução nº 30	t1000n8u2.txt	124	3.1660	138	2.3522

Fonte: Elaborada pelo autor (2026).

Tabela 33 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n8u4.txt	412	43.2210	387	7.1567
Execução nº 2	t1000n8u4.txt	386	22.6354	395	6.8747
Execução nº 3	t1000n8u4.txt	401	19.5448	412	6.9381
Execução nº 4	t1000n8u4.txt	421	10.3806	383	7.1392
Execução nº 5	t1000n8u4.txt	417	31.7911	362	7.0145
Execução nº 6	t1000n8u4.txt	420	17.8220	372	6.8751
Execução nº 7	t1000n8u4.txt	378	47.4204	422	6.8796
Execução nº 8	t1000n8u4.txt	405	14.0130	393	7.1259
Execução nº 9	t1000n8u4.txt	430	16.5837	405	6.7425
Execução nº 10	t1000n8u4.txt	419	15.2489	406	7.2031
Execução nº 11	t1000n8u4.txt	433	27.0837	398	6.8888
Execução nº 12	t1000n8u4.txt	425	38.5900	384	7.0803
Execução nº 13	t1000n8u4.txt	411	24.4247	417	6.8574
Execução nº 14	t1000n8u4.txt	425	13.4195	380	6.9699
Execução nº 15	t1000n8u4.txt	399	28.6579	370	7.0462
Execução nº 16	t1000n8u4.txt	410	24.7243	393	6.9214
Execução nº 17	t1000n8u4.txt	376	23.1395	385	7.1229
Execução nº 18	t1000n8u4.txt	401	19.7370	370	7.1214
Execução nº 19	t1000n8u4.txt	433	27.1830	373	6.9385
Execução nº 20	t1000n8u4.txt	397	18.2949	397	6.9371
Execução nº 21	t1000n8u4.txt	391	35.1656	405	7.0003
Execução nº 22	t1000n8u4.txt	430	19.3530	393	6.9533
Execução nº 23	t1000n8u4.txt	407	23.7498	402	6.9992
Execução nº 24	t1000n8u4.txt	426	20.4280	415	7.0153
Execução nº 25	t1000n8u4.txt	394	17.8491	377	7.1562
Execução nº 26	t1000n8u4.txt	403	26.1966	374	6.9842
Execução nº 27	t1000n8u4.txt	430	37.7132	378	7.1590
Execução nº 28	t1000n8u4.txt	399	17.6496	366	7.2976
Execução nº 29	t1000n8u4.txt	401	40.3813	407	6.9953
Execução nº 30	t1000n8u4.txt	420	23.2163	374	7.0942

Fonte: Elaborada pelo autor (2026).

Tabela 34 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n8u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n8u6.txt	763	138.8289	689	13.1500
Execução nº 2	t1000n8u6.txt	719	55.4487	718	12.7437
Execução nº 3	t1000n8u6.txt	743	111.5826	687	12.8676
Execução nº 4	t1000n8u6.txt	759	59.3041	673	13.3411
Execução nº 5	t1000n8u6.txt	725	80.1388	724	12.8227
Execução nº 6	t1000n8u6.txt	724	113.2461	661	12.9159
Execução nº 7	t1000n8u6.txt	742	111.8035	714	13.4847
Execução nº 8	t1000n8u6.txt	732	77.5958	696	13.3721
Execução nº 9	t1000n8u6.txt	741	59.2291	704	13.4185
Execução nº 10	t1000n8u6.txt	751	182.3349	711	13.1307
Execução nº 11	t1000n8u6.txt	758	75.8292	712	13.3675
Execução nº 12	t1000n8u6.txt	750	102.9947	675	13.3200
Execução nº 13	t1000n8u6.txt	757	102.0162	707	13.7941
Execução nº 14	t1000n8u6.txt	658	62.5309	700	12.6565
Execução nº 15	t1000n8u6.txt	688	110.8136	705	13.1386
Execução nº 16	t1000n8u6.txt	708	95.0768	712	13.3727
Execução nº 17	t1000n8u6.txt	740	110.9112	667	13.6862
Execução nº 18	t1000n8u6.txt	753	149.6319	691	13.5259
Execução nº 19	t1000n8u6.txt	702	66.4835	671	13.2148
Execução nº 20	t1000n8u6.txt	725	108.7672	709	14.1069
Execução nº 21	t1000n8u6.txt	722	124.5094	694	13.0592
Execução nº 22	t1000n8u6.txt	751	70.01270	687	13.4034
Execução nº 23	t1000n8u6.txt	753	113.9994	647	12.7754
Execução nº 24	t1000n8u6.txt	746	82.6413	668	13.2319
Execução nº 25	t1000n8u6.txt	625	61.3732	674	13.4643
Execução nº 26	t1000n8u6.txt	726	88.6520	712	12.8525
Execução nº 27	t1000n8u6.txt	778	95.3481	657	13.2157
Execução nº 28	t1000n8u6.txt	754	115.5896	718	13.0948
Execução nº 29	t1000n8u6.txt	739	145.2919	680	13.8118
Execução nº 30	t1000n8u6.txt	686	74.5856	674	13.2433

Fonte: Elaborada pelo autor (2026).

Tabela 35 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u2.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n9u2.txt	138	3.7919	179	2.3885
Execução nº 2	t1000n9u2.txt	166	1.4205	169	2.2582
Execução nº 3	t1000n9u2.txt	149	2.5433	157	2.2926
Execução nº 4	t1000n9u2.txt	157	2.5129	164	2.4005
Execução nº 5	t1000n9u2.txt	183	4.1677	162	2.3065
Execução nº 6	t1000n9u2.txt	174	2.1825	166	2.2571
Execução nº 7	t1000n9u2.txt	168	3.5268	156	2.3347
Execução nº 8	t1000n9u2.txt	168	3.3544	163	2.2613
Execução nº 9	t1000n9u2.txt	170	2.0682	162	2.2404
Execução nº 10	t1000n9u2.txt	169	2.5528	163	2.3372
Execução nº 11	t1000n9u2.txt	126	1.8994	166	2.2742
Execução nº 12	t1000n9u2.txt	165	4.5925	159	2.3241
Execução nº 13	t1000n9u2.txt	176	2.9784	161	2.2760
Execução nº 14	t1000n9u2.txt	170	1.5243	156	2.3647
Execução nº 15	t1000n9u2.txt	158	1.4288	164	2.2751
Execução nº 16	t1000n9u2.txt	125	3.1099	178	2.2781
Execução nº 17	t1000n9u2.txt	158	1.4496	162	2.3394
Execução nº 18	t1000n9u2.txt	156	2.5024	178	2.3044
Execução nº 19	t1000n9u2.txt	172	3.1236	165	2.3234
Execução nº 20	t1000n9u2.txt	103	1.8765	161	2.2307
Execução nº 21	t1000n9u2.txt	91	2.2694	172	2.2452
Execução nº 22	t1000n9u2.txt	179	1.9838	168	2.3058
Execução nº 23	t1000n9u2.txt	163	1.0628	150	2.2266
Execução nº 24	t1000n9u2.txt	173	2.7127	167	2.3555
Execução nº 25	t1000n9u2.txt	156	6.3641	173	2.2886
Execução nº 26	t1000n9u2.txt	154	2.8688	169	2.2896
Execução nº 27	t1000n9u2.txt	169	4.3870	163	2.3366
Execução nº 28	t1000n9u2.txt	171	3.0296	167	2.2755
Execução nº 29	t1000n9u2.txt	181	2.2738	170	2.2858
Execução nº 30	t1000n9u2.txt	173	2.8806	162	2.2900

Fonte: Elaborada pelo autor (2026).

Tabela 36 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u4.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº 1	t1000n9u4.txt	441	33.5868	411	6.9744
Execução nº 2	t1000n9u4.txt	421	20.8785	404	6.7793
Execução nº 3	t1000n9u4.txt	431	16.1007	406	6.8903
Execução nº 4	t1000n9u4.txt	434	15.1011	401	6.8898
Execução nº 5	t1000n9u4.txt	403	17.4303	387	6.8740
Execução nº 6	t1000n9u4.txt	411	28.9958	386	6.7781
Execução nº 7	t1000n9u4.txt	393	36.0416	394	7.0367
Execução nº 8	t1000n9u4.txt	442	22.7932	403	6.8737
Execução nº 9	t1000n9u4.txt	428	21.3068	393	6.6853
Execução nº 10	t1000n9u4.txt	419	21.2890	423	6.8897
Execução nº 11	t1000n9u4.txt	442	18.7792	389	6.6223
Execução nº 12	t1000n9u4.txt	428	22.7157	406	6.6682
Execução nº 13	t1000n9u4.txt	407	22.1911	364	7.0448
Execução nº 14	t1000n9u4.txt	415	24.8774	416	6.7484
Execução nº 15	t1000n9u4.txt	408	34.5885	399	6.7047
Execução nº 16	t1000n9u4.txt	394	22.9156	383	6.8092
Execução nº 17	t1000n9u4.txt	370	19.1724	386	6.9374
Execução nº 18	t1000n9u4.txt	405	41.1980	416	7.3129
Execução nº 19	t1000n9u4.txt	376	17.6308	406	6.7953
Execução nº 20	t1000n9u4.txt	435	18.0194	402	6.7474
Execução nº 21	t1000n9u4.txt	413	22.2876	385	7.0655
Execução nº 22	t1000n9u4.txt	418	16.4709	401	6.7782
Execução nº 23	t1000n9u4.txt	425	33.6764	401	7.0902
Execução nº 24	t1000n9u4.txt	418	20.2782	384	6.8247
Execução nº 25	t1000n9u4.txt	403	17.3372	413	7.1032
Execução nº 26	t1000n9u4.txt	383	21.8834	377	6.5467
Execução nº 27	t1000n9u4.txt	394	22.2186	412	6.8580
Execução nº 28	t1000n9u4.txt	434	35.0190	404	6.8116
Execução nº 29	t1000n9u4.txt	406	39.9910	393	6.6073
Execução nº 30	t1000n9u4.txt	410	36.2152	402	6.6865

Fonte: Elaborada pelo autor (2026).

Tabela 37 – Comparação entre GRASP e Algoritmo Genético para a instância t1000n9u6.txt

Execuções	Instância	GRASP		AG	
		Solução	Tempo (s)	Solução	Tempo (s)
Execução nº1	t1000n9u6.txt	756	111.9458	690	12.4961
Execução nº2	t1000n9u6.txt	782	75.6879	692	12.8222
Execução nº3	t1000n9u6.txt	738	106.4270	742	12.2128
Execução nº4	t1000n9u6.txt	734	67.9283	711	12.4186
Execução nº5	t1000n9u6.txt	743	86.4579	697	12.6294
Execução nº6	t1000n9u6.txt	756	126.1321	717	12.6976
Execução nº7	t1000n9u6.txt	766	98.7197	733	13.1982
Execução nº8	t1000n9u6.txt	754	81.2380	754	12.9004
Execução nº9	t1000n9u6.txt	762	51.9789	715	12.4785
Execução nº10	t1000n9u6.txt	789	105.9039	732	12.6039
Execução nº11	t1000n9u6.txt	776	90.1222	736	13.0811
Execução nº12	t1000n9u6.txt	775	106.9643	719	12.6485
Execução nº13	t1000n9u6.txt	781	85.5551	719	12.4724
Execução nº14	t1000n9u6.txt	785	108.3730	733	13.1207
Execução nº15	t1000n9u6.txt	771	138.6757	708	12.6355
Execução nº16	t1000n9u6.txt	759	102.0532	727	13.4172
Execução nº17	t1000n9u6.txt	786	106.8957	683	13.9826
Execução nº18	t1000n9u6.txt	781	99.8003	717	13.5773
Execução nº19	t1000n9u6.txt	721	40.3525	700	13.6846
Execução nº20	t1000n9u6.txt	769	152.3358	750	12.9334
Execução nº21	t1000n9u6.txt	714	141.1236	695	13.8443
Execução nº22	t1000n9u6.txt	785	68.1292	702	12.6343
Execução nº23	t1000n9u6.txt	735	100.0027	692	12.3641
Execução nº24	t1000n9u6.txt	753	80.3220	721	12.7277
Execução nº25	t1000n9u6.txt	778	81.3343	713	12.8413
Execução nº26	t1000n9u6.txt	779	116.4607	700	13.1207
Execução nº27	t1000n9u6.txt	771	81.5183	679	12.9009
Execução nº28	t1000n9u6.txt	729	93.0057	710	13.1659
Execução nº29	t1000n9u6.txt	780	126.1622	715	12.4337
Execução nº30	t1000n9u6.txt	767	118.3464	724	12.7303

Fonte: Elaborada pelo autor (2026).