

Proposta de um algoritmo de busca local para minimizar o tempo para realização de manutenção em espaços públicos

Douglas Junior Vaz Cavalcante, João Antônio Mendes Oliveira, Rafael Henrique Palma Lima

Resumo: O estudo busca aplicar uma técnica de metaheurística *Hill Climbing* para o problema de manutenção de espaços públicos em uma cidade. Cada praça apresenta áreas e tempos de manutenção diferentes, com três equipes responsáveis pela limpeza de 21 locais, cada um com respectivas coordenadas no plano cartesiano, as quais podem ser alteradas para a formação de um novo problema. Considerou-se que as equipes são idênticas, seja em número de funcionários como no ritmo de corte por pessoa. Além disso, uma matriz com os tempos de viagem foi elaborada para calcular o tempo de movimentação das equipes. O problema de manutenção das praças foi modelado como um problema de sequenciamento de máquinas paralelas com tempo de setup dependente da ordem, utilizando a matriz de tempos de viagem como sendo os tempos de setup. O objetivo considerado foi a minimização do tempo total para que a manutenção fosse realizada em todos os locais, o que é equivalente a minimizar o makespan em problemas de sequenciamento. Criou-se técnicas de geração de soluções aleatórias, método de representação e análise de uma solução bem como exploração da vizinhança, de acordo com as características deste problema, de forma alcançar o objetivo. O algoritmo proposto obteve resultados satisfatórios, obtendo redução considerável no *makespan*, além de um nivelamento da distribuição das tarefas entre as equipes.

Palavras chave: *Hill Climbing*; Sequenciamento de Produção; Máquinas Idênticas; Pesquisa Operacional

Proposal local search algorithm to minimize time for maintenance in public spaces

Abstract: The study seeks to apply a Hill Climbing metaheuristic technique to the problem of maintaining public spaces in a city. Each square has different areas and maintaining times, with three teams responsible for cleaning 21 sites, each with their own coordinates in the cartesian plane, which can be changed to form a new problem. Teams were found to be identical, both in number of employees and in cut rate per person. In addition, a matrix of travel times was created to calculate the movement time of the teams. The squares maintenance problem was modeled as a parallel machine sequencing problem with order dependent setup time, using the travel time matrix as the setup times. The objective considered was to minimize the total time for maintenance to be performed at all locations, which is equivalent to minimizing makespan in sequencing problems. Techniques of random solution generation, method of representation and analysis of a solution as well as exploration of the neighborhood, according to the characteristics of this problem, were created in order to reach the objective. The proposed algorithm obtained satisfactory results, obtaining considerable reduction in makespan, as well as leveling the distribution of tasks among teams.

Key-words: Hill Climbing; Production Sequencing; Identical machines; Operational Research

1. Introdução

A limpeza e manutenção de locais públicos sempre foi uma preocupação para a gestão municipal. O objetivo desse estudo é elaborar um algoritmo capaz de minimizar o tempo de manutenção de praças públicas de uma determinada cidade, através da roteirização de visitas, por meio de uma heurística Hill Climbing.

Para o estudo, serão considerados 21 locais e 3 equipes, disponíveis para realização do serviço, todos os dados utilizados para a elaboração do algoritmo são fictícios. As equipes são idênticas e têm um

expediente de 8 horas diárias, tendo como função, visitar e limpar todas as praças ao longo de uma semana de trabalho. Para fins práticos, será considerado que o desempenho de cada equipe seja idêntico as demais.

O objetivo do algoritmo elaborado é minimizar o tempo de manutenção e limpeza das praças relativo a todas às equipes, de modo que a solução encontrada divida o trabalho de forma balanceada entre as equipes, buscando sempre a minimização do tempo gasto com a execução do serviço.

Para medir a qualidade das soluções encontradas, será utilizado o maior tempo de execução entre equipes, denominado makespan, ou seja, quanto menor o makespan, melhor será a solução. É possível classificar o problema apresentado como sequenciamento de sequenciamento de máquinas paralelas com tempo de setup dependente, portanto, o algoritmo pode ser utilizado para problemas semelhantes.

2. Referencial Teórico

2.1. Limpeza de áreas verdes e praças públicas

Neema e Ohgai (2012) falam sobre a importância de áreas verdes nas cidades, já que as mesmas possuem diversos objetivos, sendo espaços de recreações, prática de atividades físicas e pontos de encontro, onde moradores da região possam desfrutar do local para lazer. Além disso, um espaço verde na cidade melhora a paisagem da mesma, também é notada melhoria na qualidade do ar, queda na temperatura e redução de ruídos.

Logo as áreas verdes de uma cidade podem ser consideradas como um investimento, como é defendido por Maland (2012), já que essas áreas verdes geram retorno, mesmo que muitas vezes só é visto os custos iniciais, que segundo o autor, podem ser: a aquisição da área, o desenvolvimento e custos de oportunidades.

Já quanto a manutenção, os serviços para manter o local limpo agradável podem ser variados, tendo em vista diferentes equipes para fazer a manutenção, como uma equipe para podar árvores, outra para limpeza, recolher lixos e aparar grama. (VALADE; KAHN; CHOEPHEL, 2013).

2.2. Meta-Heurística

Por se tratar de problemas complexos, muitas vezes é necessário, e mais viável, a utilização de métodos de otimização de problemas. Onde podemos destacar os métodos que buscam por uma solução ótima, garantindo que seja a melhor, como Branch-and-bound, porém, é uma alternativa mais cara e demorada. Outra opção é a utilização de heurísticas, que geram bons resultados e requerem menos tempo e investimento (ROTHLAUF, 1998)

As meta-heurísticas em muitos casos são mais vantajosas, por motivos como ser inviável procurar por a solução ótima, por questão de tempo e custos. E também, pela meta-heurística proporcionar soluções boas o suficiente (TALBI, 2009).

Segundo Glover e Knochenberger (2003) hoje em dia as meta-heurísticas são usadas com mais frequência, porém, apesar de simples, eles salientam que as mesmas devem ser eficazes na resolução de problemas, já que as mesmas são constantemente criadas para lidar com problemas individuais para uma situação, ou seja, não sendo algo generalizado.

2.3. Hill-Climbing

De acordo com Yuan e Zhang (2015), o *Hill Climbing* é uma metodologia de algoritmo que busca uma melhoria contínua através da exploração da vizinhança até que se atinja o pico mais alto, ou seja, a cada iteração o algoritmo é capaz de explorar uma determinada vizinhança, selecionando o melhor vizinho e assim sucessivamente até que não se haja mais melhoria.

Deste modo, Russel (2003), faz alusão a escalada de uma montanha, onde o algoritmo, passo a passo, sobe mais um degrau, até que se atinja o topo, onde não há mais melhoria aparente. Quando se chega neste ponto, o algoritmo se reinicia a partir da geração de uma nova geração de solução aleatória.

2.4. Vizinho Mais Próximo

Para Fernandes (2005), o algoritmo do Vizinho Mais Próximo é uma técnica muito simples de ser aplicada, que é capaz de resolver um problema através da distância entre os locais que estão sendo analisados no problema.

Segundo Bricce (2016) é o Vizinho Mais Próximo uma técnica capaz de realizar a comparação entre as distancias dos vizinho e escolher o imediatamente mais próximo. Russel (2003), explica que a cada iteração o algoritmo busca seu vizinho imediatamente mais próximo, selecionando-o, a partir deste vizinho selecionado, o algoritmo deve ser capaz de visualizar os vizinhos que ainda não foram visitados, escolhendo o de menor distância. Tal procedimento é repetido n vezes até que todos os pontos tenham sido visitados e o ciclo se fecha.

3. Análise do Processo

3.1. O processo de limpeza

O processo trata da manutenção em praças publicas de uma cidade, no qual haveriam 21 praças e 3 equipes disponíveis para realização do serviço. As equipes são idênticas e têm um expediente de 8 horas diárias, partiriam da última praça limpa, retomando a mesma no outro dia, caso não seja possível concluir em um dia (8 horas). Assim, cada praça teria sua matriz distancia para todos os pontos.

3.2. O algoritmo

A partir de uma solução inicial aleatória gerada na forma de um vetor, ocorre a avaliação do *makespan*, que consiste na somatória de todos os tempos de deslocamento mais o tempo de limpeza das praças de cada equipe levando em consideração ainda que a solução deverá resultar makespans menores que 2880 e maiores que 2000 (limite superior e inferior na tomada de decisões) para que a solução inicial seja aceita, garantindo uma solução inicial razoável para o problema, além de garantir que seja respeitada a carga horária de trabalho semanal das equipes.

Em seguida, são formados 6 vizinhos a partir da mudança de praças entre equipes – utilizando os maiores tempos de corte de grama para cada equipe – e, posteriormente, usa-se os menores tempos para gerar outros 6 vizinhos. Assim essas soluções também são avaliadas e escolhe-se a de menor *makespan* como candidata.

Escolhida a solução candidata, será gerada uma nova vizinhança a partir desta solução, até que se encontre o ótimo local para o vetor aleatório inicial. Por fim, o processo se iniciará, por N vezes, até que seja encontrada a melhor solução global para o problema.

4. Dados do Problema

Inicialmente foram listadas 21 praças com coordenadas cartesianas geradas aleatoriamente, a partir destes pontos, foi construída a tabela utilizando-se a distância euclidiana para calcular a distância entre todas as praças, Tabela 1:

Tabela 1 - Matriz de Distância (De / Para - Simétrico) (Fonte: O Autor)

Matriz de Distâncias (De / Para - Simétrico)																					
De/Para	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	412	706	1093	310	1007	452	814	758	446	444	983	501	213	766	475	702	554	670	693	1077

2	412	0	838	1104	722	823	864	1055	561	567	722	911	167	542	474	887	1046	507	1051	623	727
3	706	838	0	433	775	664	841	301	1376	1151	334	462	757	517	745	796	536	1229	666	371	1043
4	1093	1104	433	0	1203	522	1274	618	1665	1525	766	259	973	929	813	1228	927	1566	1074	484	1007
5	310	722	775	1203	0	1236	142	763	984	556	444	1155	806	302	1049	175	536	738	444	895	1371
6	1007	823	664	522	1236	0	1350	958	1351	1359	893	268	658	934	384	1328	1185	1329	1293	348	489
7	452	864	841	1274	142	1350	0	778	1102	653	508	1250	946	419	1182	71	505	849	379	1005	1507
8	814	1055	301	618	763	958	778	0	1551	1246	370	733	1006	601	1037	715	326	1367	484	671	1342
9	758	561	1376	1665	984	1351	1102	1551	0	477	1187	1468	706	964	973	1155	1459	263	1411	1184	1110
10	446	567	1151	1525	556	1359	653	1246	477	0	879	1388	729	650	1038	714	1079	214	1000	1089	1287
11	444	722	334	766	444	893	508	370	1187	879	0	754	709	232	830	466	343	997	409	547	1160
12	983	911	462	259	1155	268	1250	733	1468	1388	754	0	764	858	566	1216	998	1399	1125	299	749
13	501	167	757	973	806	658	946	1006	706	729	709	764	0	574	309	958	1049	673	1078	489	584
14	213	542	517	929	302	934	419	601	964	650	232	858	574	0	772	409	510	766	510	593	1102
15	766	474	745	813	1049	384	1182	1037	973	1038	830	566	309	772	0	1180	1168	975	1235	383	332
16	475	887	796	1228	175	1328	71	715	1155	714	466	1216	958	409	1180	0	435	905	308	981	1508
17	702	1046	536	927	536	1185	505	326	1459	1079	343	998	1049	510	1168	435	0	1238	159	854	1496
18	554	507	1229	1566	738	1329	849	1367	263	214	997	1399	673	766	975	905	1238	0	1176	1102	1182
19	670	1051	666	1074	444	1293	379	484	1411	1000	409	1125	1078	510	1235	308	159	1176	0	952	1566
20	693	623	371	484	895	348	1005	671	1184	1089	547	299	489	593	383	981	854	1102	952	0	672
21	1077	727	1043	1007	1371	489	1507	1342	1110	1287	1160	749	584	1102	332	1508	1496	1182	1566	672	0

A partir dessas distâncias foi construída a tabela que mostra o tempo de deslocamento de uma praça à outra (em minutos), utilizando uma velocidade média de 40km/h, Tabela 2:

Tabela 2 - Tempos de deslocamento entre praças (min) (Fonte: O Autor)

Tempos de deslocamento entre praças (min)																					
De / Para	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0,0	6,2	10,6	16,4	4,7	15,1	6,8	12,2	11,4	6,7	6,7	14,7	7,5	3,2	11,5	7,1	10,5	8,3	10,1	10,4	16,2
2	6,2	0,0	12,6	16,6	10,8	12,3	13,0	15,8	8,4	8,5	10,8	13,7	2,5	8,1	7,1	13,3	15,7	7,6	15,8	9,3	10,9
3	10,6	12,6	0,0	6,5	11,6	10,0	12,6	4,5	20,6	17,3	5,0	6,9	11,4	7,8	11,2	11,9	8,0	18,4	10,0	5,6	15,6
4	16,4	16,6	6,5	0,0	18,1	7,8	19,1	9,3	25,0	22,9	11,5	3,9	14,6	13,9	12,2	18,4	13,9	23,5	16,1	7,3	15,1
5	4,7	10,8	11,6	18,1	0,0	18,5	2,1	11,4	14,8	8,3	6,7	17,3	12,1	4,5	15,7	2,6	8,0	11,1	6,7	13,4	20,6
6	15,1	12,3	10,0	7,8	18,5	0,0	20,2	14,4	20,3	20,4	13,4	4,0	9,9	14,0	5,8	19,9	17,8	19,9	19,4	5,2	7,3
7	6,8	13,0	12,6	19,1	2,1	20,2	0,0	11,7	16,5	9,8	7,6	18,7	14,2	6,3	17,7	1,1	7,6	12,7	5,7	15,1	22,6
8	12,2	15,8	4,5	9,3	11,4	14,4	11,7	0,0	23,3	18,7	5,5	11,0	15,1	9,0	15,6	10,7	4,9	20,5	7,3	10,1	20,1
9	11,4	8,4	20,6	25,0	14,8	20,3	16,5	23,3	0,0	7,1	17,8	22,0	10,6	14,5	14,6	17,3	21,9	3,9	21,2	17,8	16,6
10	6,7	8,5	17,3	22,9	8,3	20,4	9,8	18,7	7,1	0,0	13,2	20,8	10,9	9,8	15,6	10,7	16,2	3,2	15,0	16,3	19,3
11	6,7	10,8	5,0	11,5	6,7	13,4	7,6	5,5	17,8	13,2	0,0	11,3	10,6	3,5	12,5	7,0	5,1	15,0	6,1	8,2	17,4
12	14,7	13,7	6,9	3,9	17,3	4,0	18,7	11,0	22,0	20,8	11,3	0,0	11,5	12,9	8,5	18,2	15,0	21,0	16,9	4,5	11,2
13	7,5	2,5	11,4	14,6	12,1	9,9	14,2	15,1	10,6	10,9	10,6	11,5	0,0	8,6	4,6	14,4	15,7	10,1	16,2	7,3	8,8
14	3,2	8,1	7,8	13,9	4,5	14,0	6,3	9,0	14,5	9,8	3,5	12,9	8,6	0,0	11,6	6,1	7,7	11,5	7,6	8,9	16,5
15	11,5	7,1	11,2	12,2	15,7	5,8	17,7	15,6	14,6	15,6	12,5	8,5	4,6	11,6	0,0	17,7	17,5	14,6	18,5	5,7	5,0

16	7,1	13,3	11,9	18,4	2,6	19,9	1,1	10,7	17,3	10,7	7,0	18,2	14,4	6,1	17,7	0,0	6,5	13,6	4,6	14,7	22,6
17	10,5	15,7	8,0	13,9	8,0	17,8	7,6	4,9	21,9	16,2	5,1	15,0	15,7	7,7	17,5	6,5	0,0	18,6	2,4	12,8	22,4
18	8,3	7,6	18,4	23,5	11,1	19,9	12,7	20,5	3,9	3,2	15,0	21,0	10,1	11,5	14,6	13,6	18,6	0,0	17,6	16,5	17,7
19	10,1	15,8	10,0	16,1	6,7	19,4	5,7	7,3	21,2	15,0	6,1	16,9	16,2	7,6	18,5	4,6	2,4	17,6	0,0	14,3	23,5
20	10,4	9,3	5,6	7,3	13,4	5,2	15,1	10,1	17,8	16,3	8,2	4,5	7,3	8,9	5,7	14,7	12,8	16,5	14,3	0,0	10,1
21	16,2	10,9	15,6	15,1	20,6	7,3	22,6	20,1	16,6	19,3	17,4	11,2	8,8	16,5	5,0	22,6	22,4	17,7	23,5	10,1	0,0

Em seguida utilizou-se a ferramenta *Google Earth* para estimar as áreas das praças, cada equipe teria 6 trabalhadores que limpariam em média 2m² por minuto cada, totalizando 12m² por minuto por equipe. Assim chegamos ao tempo de processamento que cada praça teria para sua manutenção (Tabela 3).

Tabela 3: Área das praças a serem visitadas (Fonte: O Autor)

Praça	Área (m ²)	Tempo (min)
1	6412	534
2	2040,5	170
3	1719,2	143
4	3466,4	289
5	4343,5	362
6	2088,8	174
7	3692,5	308
8	3738	312
9	2338	195
10	7609,7	634
11	12040	1003
12	1467,9	122
13	3933,3	328
14	1410,5	118
15	1409,1	117
16	1509,9	1256
17	2461,9	205
18	6125,7	510
19	3523,8	294
20	1400	117
21	1082,9	90,242

Para facilitar a análise e implementação do algoritmo, foi elaborada uma tabela que resulta o tempo de deslocamento entre as praças e o tempo para a praça ter sua manutenção realizada. Ou seja, o tempo de locomoção relativo à distância entre as praças foi somado com o tempo necessário para realizar a limpeza e manutenção da praça de partida, conforme apresentado na Tabela 4:

Tabela 4 - Tempo de Deslocamento + Limpeza (Fonte: O Autor)

Tempo de deslocamento + Tempo de limpeza																					
De / Para	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	534,0	540,2	544,6	550,4	538,7	549,1	540,8	546,2	545,4	540,7	540,7	548,7	541,5	537,2	545,5	541,1	544,5	542,3	544,1	544,4	550,2
2	176,2	170,0	182,6	186,6	180,8	182,3	183,0	185,8	178,4	178,5	180,8	183,7	172,5	178,1	177,1	183,3	185,7	177,6	185,8	179,3	180,9
3	153,6	155,6	143,0	149,5	154,6	153,0	155,6	147,5	163,6	160,3	148,0	149,9	154,4	150,8	154,2	154,9	151,0	161,4	153,0	148,6	158,6
4	305,4	305,6	295,5	289,0	307,1	296,8	308,1	298,3	314,0	311,9	300,5	292,9	303,6	302,9	301,2	307,4	302,9	312,5	305,1	296,3	304,1
5	366,7	372,8	373,6	380,1	362,0	380,5	364,1	373,4	376,8	370,3	368,7	379,3	374,1	366,5	377,7	364,6	370,0	373,1	368,7	375,4	382,6
6	189,1	186,3	184,0	181,8	192,5	174,0	194,2	188,4	194,3	194,4	187,4	178,0	183,9	188,0	179,8	193,9	191,8	193,9	193,4	179,2	181,3
7	314,8	321,0	320,6	327,1	310,1	328,2	308,0	319,7	324,5	317,8	315,6	326,7	322,2	314,3	325,7	309,1	315,6	320,7	313,7	323,1	330,6
8	324,2	327,8	316,5	321,3	323,4	326,4	323,7	312,0	335,3	330,7	317,5	323,0	327,1	321,0	327,6	322,7	316,9	332,5	319,3	322,1	332,1
9	206,4	203,4	215,6	220,0	209,8	215,3	211,5	218,3	195,0	202,1	212,8	217,0	205,6	209,5	209,6	212,3	216,9	198,9	216,2	212,8	211,6
10	640,7	642,5	651,3	656,9	642,3	654,4	643,8	652,7	641,1	634,0	647,2	654,8	644,9	643,8	649,6	644,7	650,2	637,2	649,0	650,3	653,3
11	1009,7	1013,8	1008,0	1014,5	1009,7	1016,4	1010,6	1008,5	1020,8	1016,2	1003,0	1014,3	1013,6	1006,5	1015,5	1010,0	1008,1	1018,0	1009,1	1011,2	1020,4
12	136,7	135,7	128,9	125,9	139,3	126,0	140,7	133,0	144,0	142,8	133,3	122,0	133,5	134,9	130,5	140,2	137,0	143,0	138,9	126,5	133,2
13	335,5	330,5	339,4	342,6	340,1	337,9	342,2	343,1	338,6	338,9	338,6	339,5	328,0	336,6	332,6	342,4	343,7	338,1	344,2	335,3	336,8
14	121,2	126,1	125,8	131,9	122,5	132,0	124,3	127,0	132,5	127,8	121,5	130,9	126,6	118,0	129,6	124,1	125,7	129,5	125,6	126,9	134,5
15	128,5	124,1	128,2	129,2	132,7	122,8	134,7	132,6	131,6	132,6	129,5	125,5	121,6	128,6	117,0	134,7	134,5	131,6	135,5	122,7	122,0
16	1263,1	1269,3	1267,9	1274,4	1258,6	1275,9	1257,1	1266,7	1273,3	1266,7	1263,0	1274,2	1270,4	1262,1	1273,7	1256,0	1262,5	1269,6	1260,6	1270,7	1278,6
17	215,5	220,7	213,0	218,9	213,0	222,8	212,6	209,9	226,9	221,2	210,1	220,0	220,7	212,7	222,5	211,5	205,0	223,6	207,4	217,8	227,4
18	518,3	517,6	528,4	533,5	521,1	529,9	522,7	530,5	513,9	513,2	525,0	531,0	520,1	521,5	524,6	523,6	528,6	510,0	527,6	526,5	527,7
19	304,1	309,8	304,0	310,1	300,7	313,4	299,7	301,3	315,2	309,0	300,1	310,9	310,2	301,6	312,5	298,6	296,4	311,6	294,0	308,3	317,5
20	127,4	126,3	122,6	124,3	130,4	122,2	132,1	127,1	134,8	133,3	125,2	121,5	124,3	125,9	122,7	131,7	129,8	133,5	131,3	117,0	127,1
21	107,2	101,9	106,6	106,1	111,6	98,3	113,6	111,1	107,6	110,3	108,4	102,2	99,8	107,5	96,0	113,6	113,4	108,7	114,5	101,1	91,0

5. Resultados e discussões

A solução será representada na forma de um vetor, onde aparecerão as praças visitadas em ordem, havendo um divisor de cores para delimitar, e facilitar a visualização, as equipes (Equipe 1: Azul, Equipe 2: Laranja e Equipe 3: Verde). Por exemplo, na tabela 5, abaixo, o vetor indica que as praças de 1 a 8 serão limpas pela equipe 1, as praças 9 a 14 pela equipe 2 e as praças 15 a 21 pela equipe 3, seguindo a ordem em que aparecem no vetor.

Tabela 5: Vetor de representação da solução (Fonte: O Autor)

Vetor	Ordem
1	1º
2	2º
3	3º
4	4º
5	5º
6	6º
7	7º
8	8º

9	1º
10	2º
11	3º
12	4º
13	5º
14	6º
15	1º
16	2º
17	3º
18	4º
19	5º
20	6º
21	7º

5.1. Método para geração de soluções aleatórias

O próximo passo é a elaboração de um método que gere soluções aleatórias. Neste problema foi utilizada a geração completamente aleatória de números (de 0 à 21) onde são inicialmente atribuídas 7 praças para cada equipe. Contudo, para que a solução inicial não seja muito ruim, estipulamos que as equipes deveriam trabalhar um tempo mínimo, de 2000 minutos, e um tempo máximo, de 2880 minutos, sendo estes, limitantes inferior e superior, respectivamente. Estes limites, além de garantir soluções iniciais razoavelmente boas, referem-se às cargas horárias semanais que as equipes devem respeitar, dessa forma as equipes visitariam as praças em um intervalo aceitável, nem tão cedo a ponto de não ter o que limpar, e nem tão tarde, a ponto de a praça necessitar de trabalhos mais específicos.

5.2. Método de avaliação da qualidade da solução

A avaliação da qualidade da solução foi desenvolvida com base na leitura e armazenamento da matriz de tempos totais na qual a partir da solução dada, o algoritmo lê o vetor e associa à cada valor na tabela seu determinado tempo.

Assim, ao somar os tempos referentes de todas as posições do vetor surge o tempo de operação acrescido ao de deslocamento por equipe. Feito isso, o maior tempo entre todas as equipes será o *makespan* da solução gerada. Portanto, o *makespan* é de fundamental importância para este problema, já que o objetivo principal é minimizar o maior tempo entre as equipes.

No exemplo abaixo (Tabela 6) é apresentado o cálculo do *makespan* da equipe 1 – por meio dos tempos totais (Tabela 4). Então, são somados os tempos de (6 para 20)+(20 para 11)+(11 para 13)+(13 para 10)+(10 para 5)+(5 para 9) e (9 para 9) para que seja contabilizado o tempo de limpeza da praça 9.

Tabela 6: Cálculo do tempo total de uma solução qualquer

Tempos de deslocamento+Tempos de limpeza							
De/Para	6	20	11	13	10	5	9
6		179,2222					
20			125,2123				
11				1013,635			
13					338,9313		
10						642,3454	
5							376,7565
9							195

O mesmo será feito para as demais equipes, o resultado pode ser visto na Tabela 8, a qual retorna também o makespan desta solução.

Tabela 7: Exemplo do cálculo do makespan da solução (Fonte: O Autor.)

	Equipe 1	Equipe 2	Equipe 3	
	6	17	3	
	20	2	7	
	11	1	19	
	13	12	4	
	10	21	15	
	5	16	8	
	9	18	14	Makespan
TOTAL	2871	2972	1652	2972

É importante ressaltar que este exemplo aleatório apenas mostra a avaliação de uma solução geral. Neste caso específico, as duas últimas equipes apresentam tempos de processamentos fora dos limites utilizados neste trabalho.

5.3. Mecanismo de exploração da vizinhança

A partir de uma solução qualquer factível, gera-se uma vizinhança com 6 vizinhos, onde cada vizinho surge da transferência da praça com maior tempo de limpeza de uma equipe para as outras. A Tabela 8 exemplifica a exploração da vizinhança: ocorre a transferência entre a praça com maior tempo de limpeza da equipe 1 (Praça 11 com 1003 min) para a equipe 2 e posteriormente para a equipe 3.

Tabela 8: Exploração da Vizinhança (Fonte: O Autor).

S. inicial	Vizinho 1					
	Equipe 1	T limpeza	Equipe 2	T limpeza	Equipe 3	T limpeza
14	14	118	4	289	3	143
13	13	328	7	308	21	91
12	12	122	15	117	16	1256
8	8	312	5	362	6	174
11	11	1003	18	510	19	294
20	20	117	17	205	9	195
10	10	634	1	534	2	170
4	-	-	11	1003	-	-
7	TOTAL	1631	-	3328	-	2323
15	Makespan	3328				
5	Vizinho 2					
18	Equipe 1	T limpeza	Equipe 2	T limpeza	Equipe 3	T limpeza
17	14	118	4	289	3	143
1	13	328	7	308	21	91
3	12	122	15	117	16	1256
21	8	312	5	362	6	174
16						
6						
19						
9						
2						

11	1003	18	510	19	294
20	117	17	205	9	195
10	634	1	534	2	170
-	-	-	-	11	1003
TOTAL	1631	-	2325	-	3326
Makespan	3326				

Através desse processo, são gerados 2 vizinhos, em seguida repete-se esse procedimento para as equipes 2 e 3, gerando então mais 2 vizinhos por procedimento, totalizando um total de 6 vizinhos. Abaixo, Tabela 9, estão representados todos os seis vizinhos gerando com suas trocas destacadas em amarelo.

Tabela 9: Vizinhaça completa e respectivos makespans (Fonte: O Autor).

Vizinho 1	Vizinho 2	Vizinho 3	Vizinho 4	Vizinho 5	Vizinho 6	
14	14	4	4	3	3	
13	13	7	7	21	21	
12	12	15	15			
8	8	5	5	6	6	
		18	18	19	19	
20	20	17	17	9	9	
10	10			2	2	
4	4	14	14	14	14	
7	7	13	13	13	13	
15	15	12	12	12	12	
5	5	8	8	8	8	
18	18	11	11	11	11	
17	17	20	20	20	20	
1	1	10	10	10	10	
11	3	1	3	16	4	
3	21	3	21	4	7	
21	16	21	16	7	15	
16	6	16	6	15	5	
6	19	6	19	5	18	
19	9	19	9	18	17	
9	2	9	2	17	1	
2	11	2	1	1	16	
Makespan	3328	3326	3168	2857	3890	3581

Caso esse processo gere algum vizinho melhor que a solução inicial, então o algoritmo adota esta como solução candidata, caso contrário a solução inicial continua sendo a atual e o algoritmo para (ótimo local).

Posteriormente, realiza-se a próxima geração de vizinhaça partindo da nova solução candidata. Faz-se a transferência de praças entre as equipes como na primeira vizinhaça, mas dessa vez utilizando

as praças com menor tempo de limpeza nas transferências. A Tabela 10 exemplifica a reorganização das praças entre as equipes gerando outros 6 vizinhos, desta vez com a troca das praças de menor tempo entre as equipes.

Tabela 10: Vizinhos gerados pela troca das praças de menores tempos (Fonte: O Autor).

S. ATUAL	Vizinho 1	Vizinho 2	Vizinho 3	Vizinho 4	Vizinho 5	Vizinho 6	
14	14	14	4	4	3	3	
13	13	13	7	7	21	21	
12	12	12	15	15	16	16	
8	8	8	5	5	6	6	
11	11	11	18	18	19	19	
20			17	17	9	9	
10	10	10	1	1	2	2	
4	4	4	14	14	14	14	
7	7	7	13	13	13	13	
15	15	15	12	12	12	12	
5	5	5	8	8	8	8	
18	18	18	11	11	11	11	
17	17	17	20	20	20	20	
1	1	1	10	10	10	10	
3	20	3	15	3	21	4	
21	3	21	3	21	4	7	
16	21	16	21	16	7	15	
6	16	6	16	6	15	5	
19	6	19	6	19	5	18	
9	19	9	19	9	18	17	
2	9	2	9	2	17	1	
	2	20	2	15	1	21	
Makespan	2695,156	2517	2517	2751	2634	2725	2634

Esse processo tem por finalidade obter grandes melhoras em soluções iniciais ruins com poucas iterações, por meio da realocação das praças com maior tempo de limpeza e, em seguida, conseguir um ajuste mais fino por meio da realocação utilizando os menores tempos de limpeza. Assim, a última etapa da geração da vizinhança será a aplicação do método do vizinho mais próximo (VMP) por equipe, para tentar determinar possíveis melhorias internas por equipe, de forma a encontrar um melhor sequenciamento da solução analisada.

Na Tabela 11, abaixo, é possível visualizar as soluções geradas, comparando-as com a solução aleatória inicial, onde pode ser observada uma redução no makespan de 2695,16 para 2501,89, o que se traduz numa redução total de 7,171%.

Tabela 11: Resultados finais (Fonte: O Autor).

Solução Inicial Gerada			Resultado pós iterações			Após VMP		
EQUIPE 1	EQUIPE 2	EQUIPE 3	EQUIPE 1	EQUIPE 2	EQUIPE 3	EQUIPE 1	EQUIPE 2	EQUIPE 3
14	4	3	13	4	3	11	7	16
13	7	21	8	7	21	8	5	19
12	15	16	11	15	16	20	14	3
8	5	6	20	5	6	13	1	12
11	18	19	10	18	19	10	18	6
20	17	9		17	9		15	21
10	1	2		1	2		4	2
	17	9		14	12		17	9
2695,15	2417,74	2430,17	2439,19	2538,933	2565,797	2427,87	2501,89	2497,23

6. Conclusão

Ao aplicar o algoritmo em uma solução simulada verificou-se que houve melhoria na distribuição de função quando comparado à solução inicial em poucas iterações. Com isso, conseguiu-se demonstrar a eficácia do algoritmo, pois além de nivelar a distribuição das tarefas entre as equipes, também mostra uma melhora do tempo total ao aplicarmos o VPM. Nessa simulação, foram consideradas equipes homogêneas, fator que facilitou a representação da matriz de tempos totais e implementação da geração da vizinhança, caso fosse aplicado na realidade, novos parâmetros deveriam ser implementados ao algoritmo.

Com os resultados encontrados, o algoritmo proposto mostrou ser eficaz, sendo capaz de resolver e encontrar soluções razoavelmente boas para problemas de sequenciamento de produção em máquinas idênticas, com tempo de setup diferentes, atingindo soluções ótimas locais com poucas iterações. Para uma melhor análise estatística da qualidade do algoritmo, seriam necessários estudos mais aprofundados, assim como a modelagem do algoritmo utilizando uma linguagem de programação.

Referências

BRICCE, Bruno Roberto et al. APLICAÇÃO DO ALGORITMO KNN PARA CONTROLE DE MOVIMENTOS DE NPCS EM UM AMBIENTE DINÂMICO (JOGO). REGRAD - Revista Eletrônica de Graduação do UNIVEM - ISSN 1984-7866, [S.l.], v. 9, n. 1, p. 18-32, aug. 2016. ISSN 1984-7866. Disponível em: <<https://revista.univem.edu.br/REGRAD/article/view/1317>>. Acesso em: 18 oct. 2019.

FERNANDES, A. M. da R. Inteligência Artificial: noções gerais. 2. ed. Visualbooks Florianópolis, 2005.

GRIFFIN-VALADE, Lavonne; KAHN, Drummond. CHOEPHEL, Tenzin. **Portland Parks and Recreation**. 2013. Disponível em: <<https://www.portlandoregon.gov/auditservices/article/459334/>>

MALAND, Myles M. **The return on investment of parks and open space: How Economics can inform design**. 2012. Disponível em: <https://getd.libs.uga.edu/pdfs/maland_myles_m_201208_mla.pdf>

NEEMA, Meher Nigar; OHGAI, Akira. **Multitype Green-Space Modeling for Urban Planning Using GA and GIS. Environment And Planning B: Planning and Design**, [s.l.], v. 40, n. 3, p.447-473, jan. 2013. SAGE Publications. <http://dx.doi.org/10.1068/b38003>.

RUSSEL, S.J. e NORVIG, P. (2003). Artificial Intelligence: A modern Approach, 2003. 2o. Ed. Prentice Hall, Upper Saddle River, NJ.

YUAN, Biao; ZHANG, Chaoyong; SHAO, Xinyu. A late acceptance hill-climbing algorithm for balancing two-sided assembly lines with multiple constraints. **Journal of Intelligent Manufacturing**, v. 26, n. 1, p. 159-168, 2015. Disponível em: <<https://link.springer.com/article/10.1007/s10845-013-0770-x>>