# Otimização Combinatória

CVRP com Simulated Annealing

César Malerba
Rodrigo A. Batista

# Sobre o Problema - VRP

- Vehicle Routing Problem
- Otimização das rotas para atender clientes em localidades diferentes e retornar ao(s) ponto(s) de partida
- Problema já definido por mais de 40 anos e considerado de difícil solução
- Foco do trabalho CVRP

# Instância do CVRP

**Instância** Um depósito único, que tem que atender demandas $c_i$, $1 \leq i \leq n$, de $n$ clientes, uma matriz de distâncias $D = (d_{ij}) \in \mathbb{R}^{(n+1)\times(n+1)}$ de distâncias entre os clientes (o índice 0 corresponde com o depósito), um número $k$ de caminhões com capacidades $C$.

**Solução** Uma partição em rotas $R_1 \cup \cdots \cup R_r$ de $[1, n]$, tal que cada rota pode ser atendido de um caminhão $\sum_{i \in R_j} c_i \leq C$, $\forall j \in [1, n]$, e uma permutação $\sigma_i$ para cada rota que define a ordem em que os clientes serem atendidos.

**Objetivo** Minimizar o custo de total de atendimento

# Sobre o Problema - CVRP

- Capacited VRP
- Veículos possuem todos a mesma capacidade
- Em todas as instâncias, há apenas 1 depósito

# Sobre a Heurística

- Simulated annealing: baseada em resfriamento controlado de cristais para diminuir os defeitos na estrutura deles
- Paramêtros essenciais: temperatura inicial e sua taxa de queda
- Optou-se por temperatura inicial fixa e taxa de queda variável

# Implementação e Testes

- Desenvolvido em C++ usando a STL
- Ambiente DevC++ e Eclipse
- Freeglut 2.2.0
- Os testes foram executados em um Pentium D805 2,66Ghz com 1GB de RAM

# CVRP com Simulated Annealing base do algoritmo

```
Solucao s = this->solucao_inicial;
while(temp > 5) {
    // busca local
    for(int i=0; i<this->n_busca_local*this->cvrp->pega_dimensao(); i++) {
        nova_sol = this->gera_nova_solucao(s);
        nova_sol.conferir_solucao();

        if(nova_sol.total_distancia() < min) {
            s_min = nova_sol;
            min = nova_sol.total_distancia();
        }

        if(nova_sol.total_distancia() <= s.total_distancia()) {
            s = nova_sol;
        } else {
            // pegar nova_sol dependendo da temperatura atual
            valor_s = s.total_distancia();
            valor_nova_sol = nova_sol.total_distancia();
            if(this->funcao_probabilidade(temp, valor_s, valor_nova_sol)) {
                s = nova_sol;
            }
        }
    }

    // baixar temperatura
    temp = this->taxa_queda_temp * temp;
}

return s_min;
```

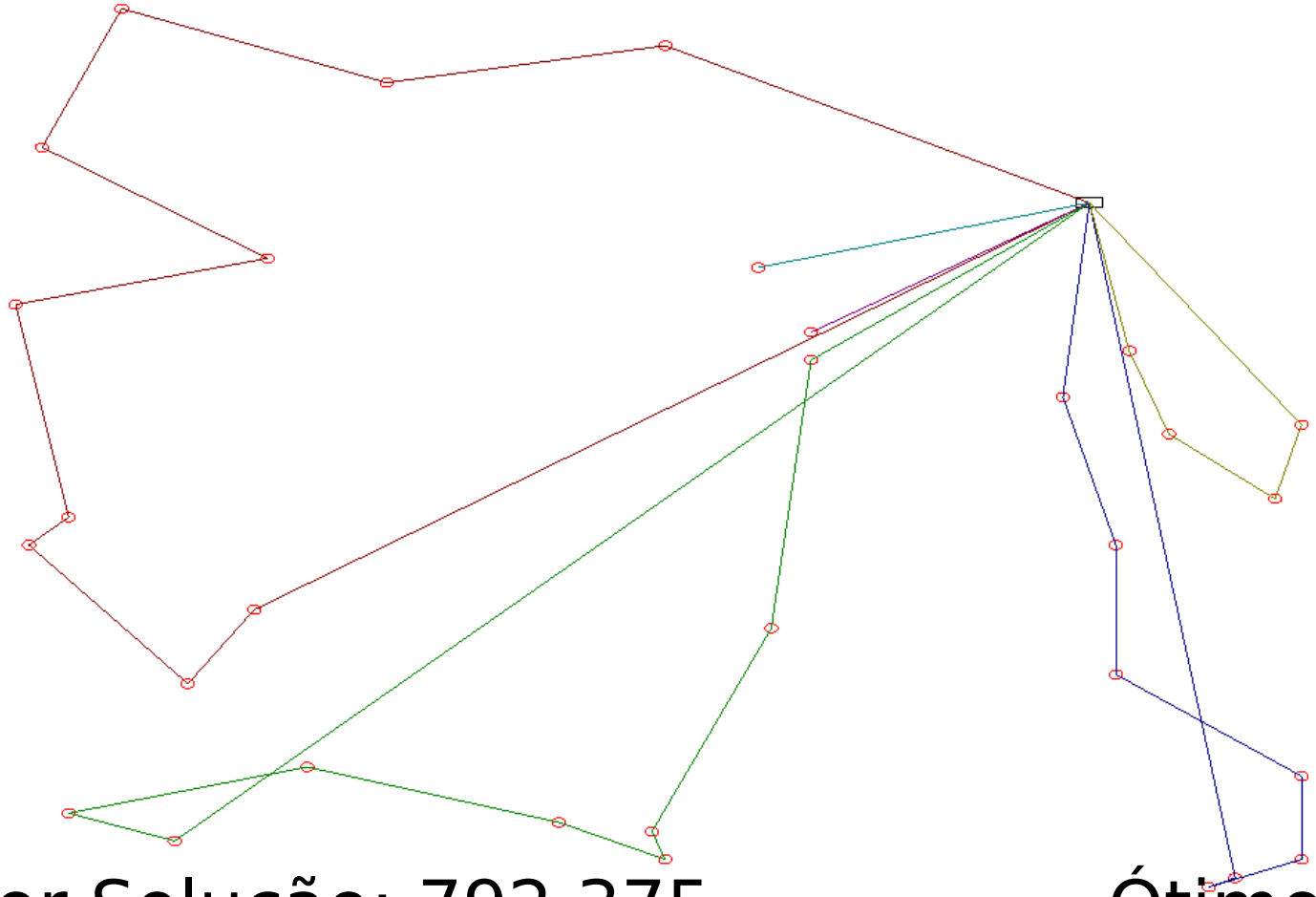
# CVRP com Simulated Annealing
## geração de vizinhança

```
int tipo_nova_rota = rand()%3;

switch(tipo_nova_rota) {
    case 0:
        nova_sol = s.trocar_cliente_entre_rotas();
        break;
    case 1:
        nova_sol = s.trocar_ordem_rota();
        break;
    case 2:
        nova_sol = s.mover_cliente_entre_rotas();
        break;
}
return nova_sol;
```

# CVRP com Simulated Annealing função de probabilidade

```
delta = 100 * (valor_nova_sol - valor_sol_atual) / valor_sol_atual;
if(delta < 1) {
    delta = 1;
}
prob = temp/delta/4.5;

if(rand()%100 < (int)prob) {
    return true;
}
return false;
```

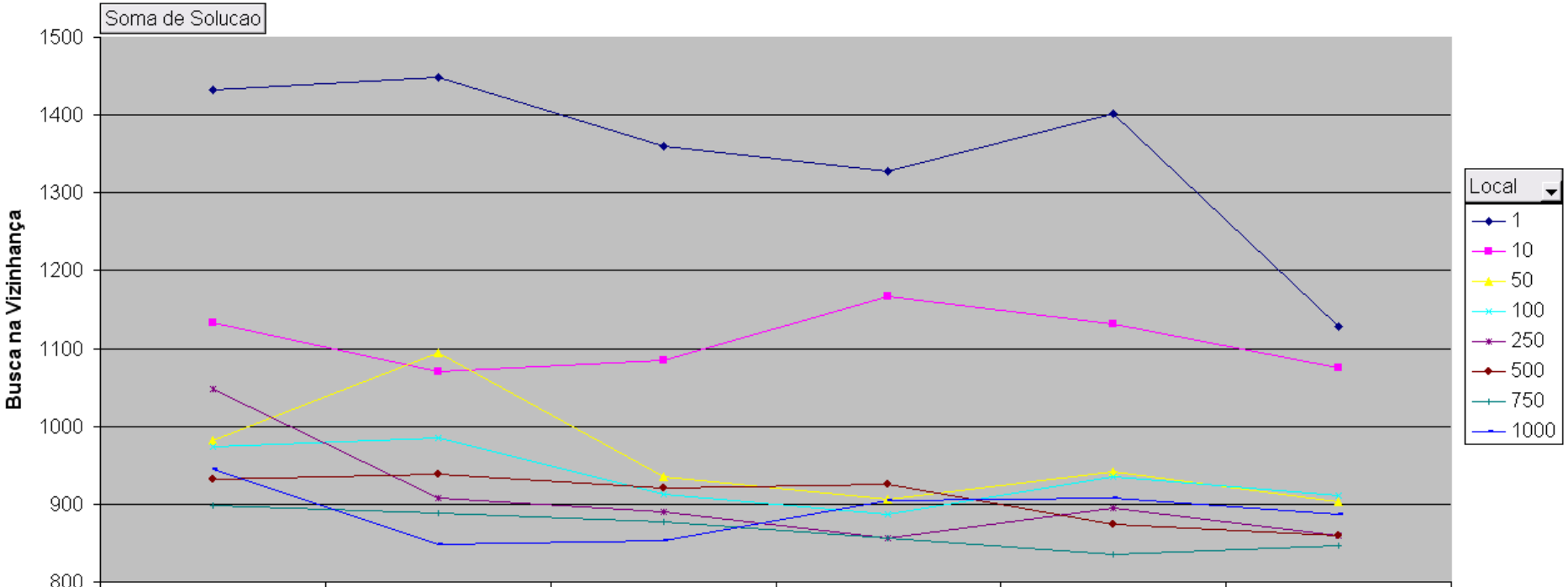
# Exemplo Execução – A-n32-k5



Melhor Solução: 792,375 Ótimo: 784

# A-n32-k5 (1)

# Ótimo: 784



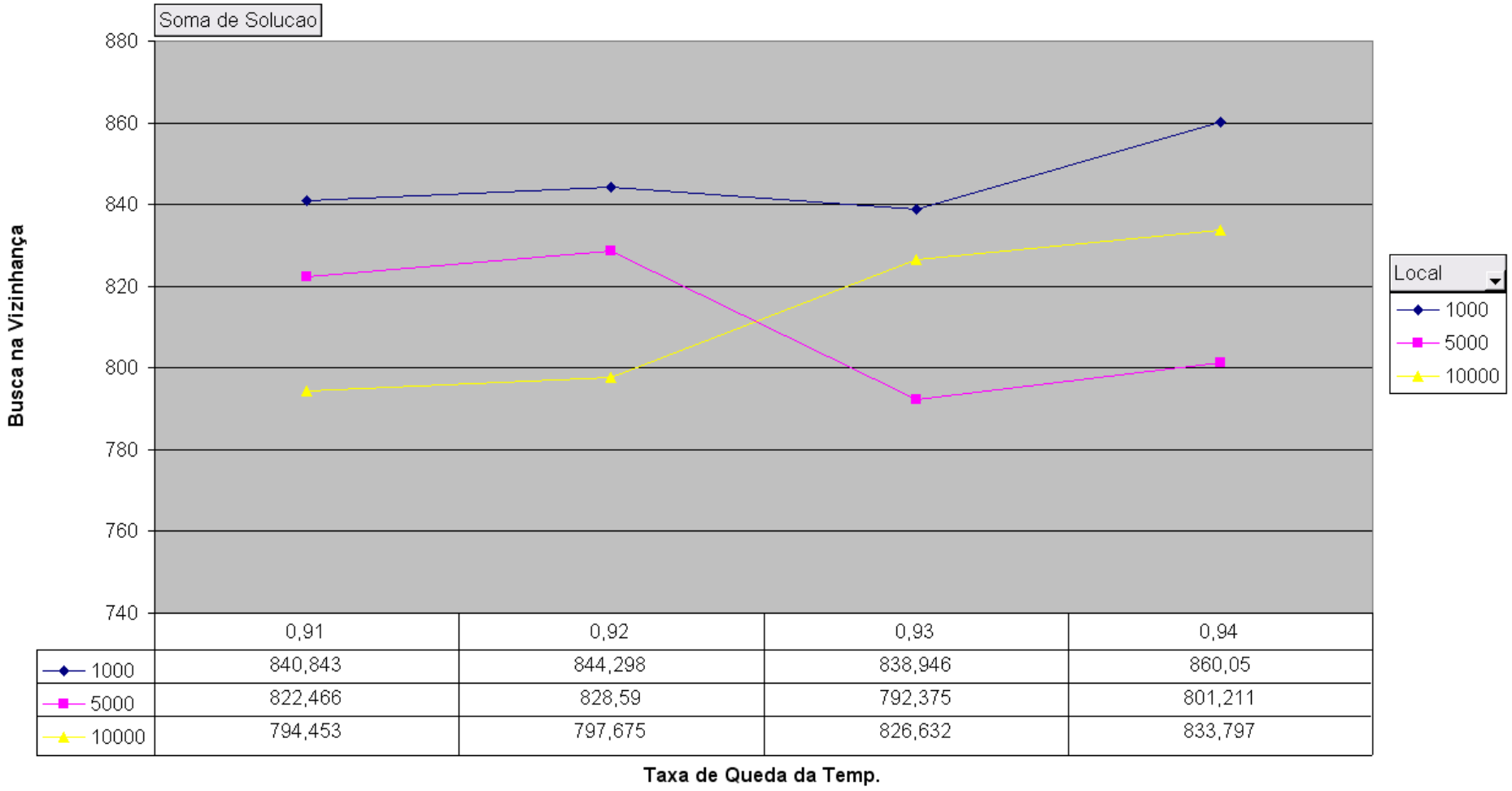

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1433,14 | 1448,55 | 1359,8 | 1327,21 | 1402,21 | 1127,86 |
| 10 | 1133,66 | 1070,05 | 1084,37 | 1167,07 | 1132,09 | 1075,33 |
| 50 | 982,476 | 1094,57 | 935,501 | 906,875 | 940,919 | 903,259 |
| 100 | 973,331 | 985,255 | 912,965 | 886,639 | 934,898 | 910,76 |
| 250 | 1048,01 | 908,43 | 890,278 | 855,921 | 894,825 | 859,329 |
| 500 | 931,554 | 938,062 | 920,208 | 925,066 | 874,131 | 860,338 |
| 750 | 898,266 | 889,148 | 876,468 | 856,541 | 835,388 | 847,429 |
| 1000 | 945,256 | 847,532 | 852,489 | 905,103 | 908,37 | 886,181 |

Taxa de Queda da Temp.

# Melhor Solução: 835,388

# A-n32-k5 (2)

# Ótimo: 784

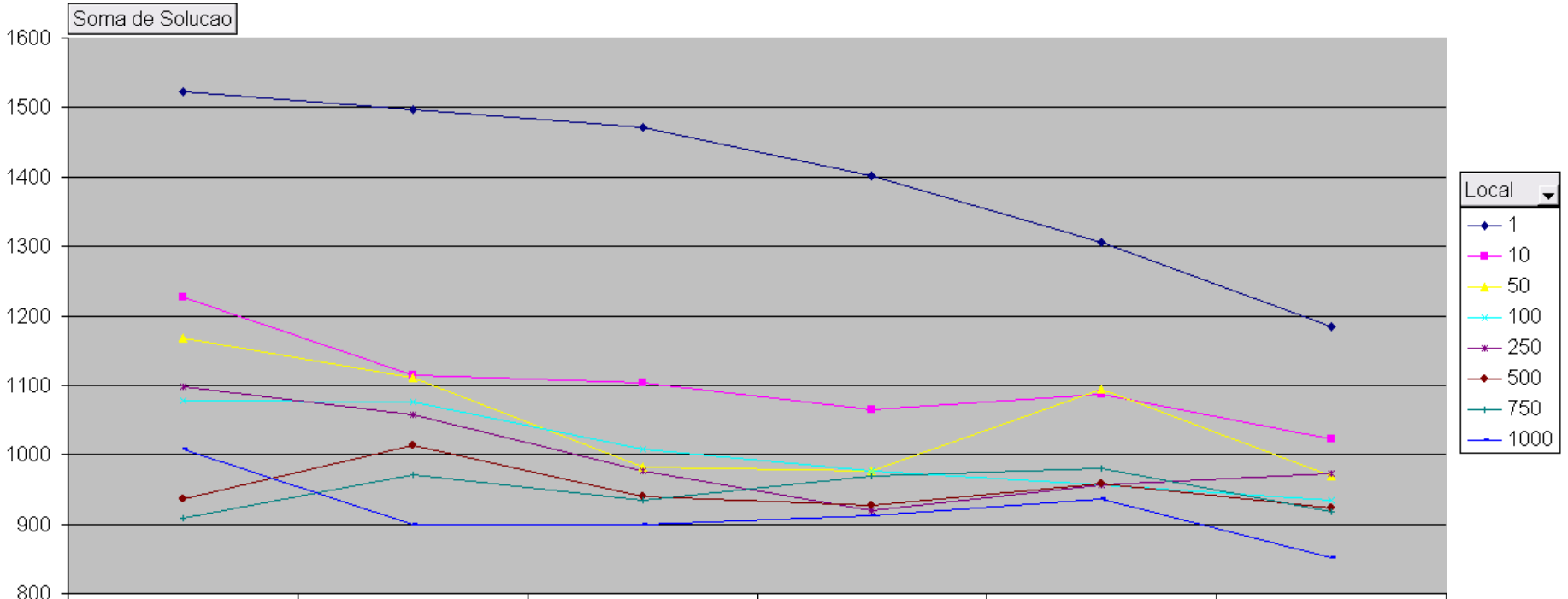



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 840,843 | 844,298 | 838,946 | 860,05 |
| 5000 | 822,466 | 828,59 | 792,375 | 801,211 |
| 10000 | 794,453 | 797,675 | 826,632 | 833,797 |

# Melhor Solução: 792,375

# A-n39-k6 (1)

# Ótimo: 831



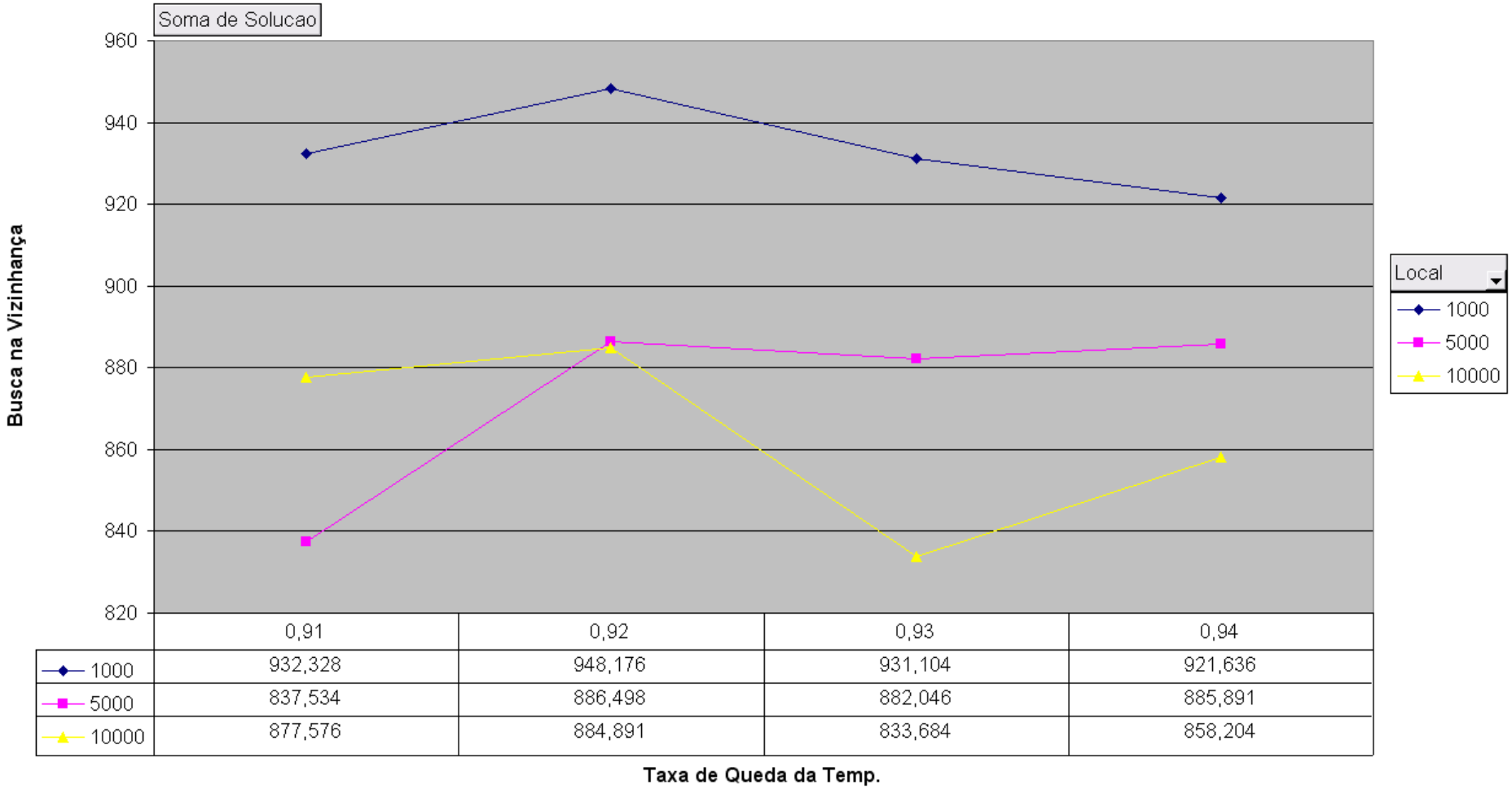

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1522,35 | 1496,37 | 1471,01 | 1400,97 | 1305,54 | 1184,71 |
| 10 | 1226,71 | 1115,07 | 1102,67 | 1064,91 | 1086 | 1022,71 |
| 50 | 1167,01 | 1110,55 | 981,699 | 976,928 | 1093,84 | 969,809 |
| 100 | 1076,8 | 1075,35 | 1008,27 | 976,535 | 955,935 | 934,815 |
| 250 | 1098,82 | 1057,72 | 976,284 | 918,953 | 956,408 | 973,326 |
| 500 | 935,554 | 1013,54 | 939,462 | 926,934 | 958,363 | 923,931 |
| 750 | 908,409 | 970,721 | 935,142 | 968,747 | 980,694 | 918,226 |
| 1000 | 1007,47 | 899,199 | 900,064 | 911,656 | 935,249 | 851,713 |

Taxa de Queda da Temp.

# Melhor Solução: 851,713

# A-n39-k6 (2)

# Ótimo: 831

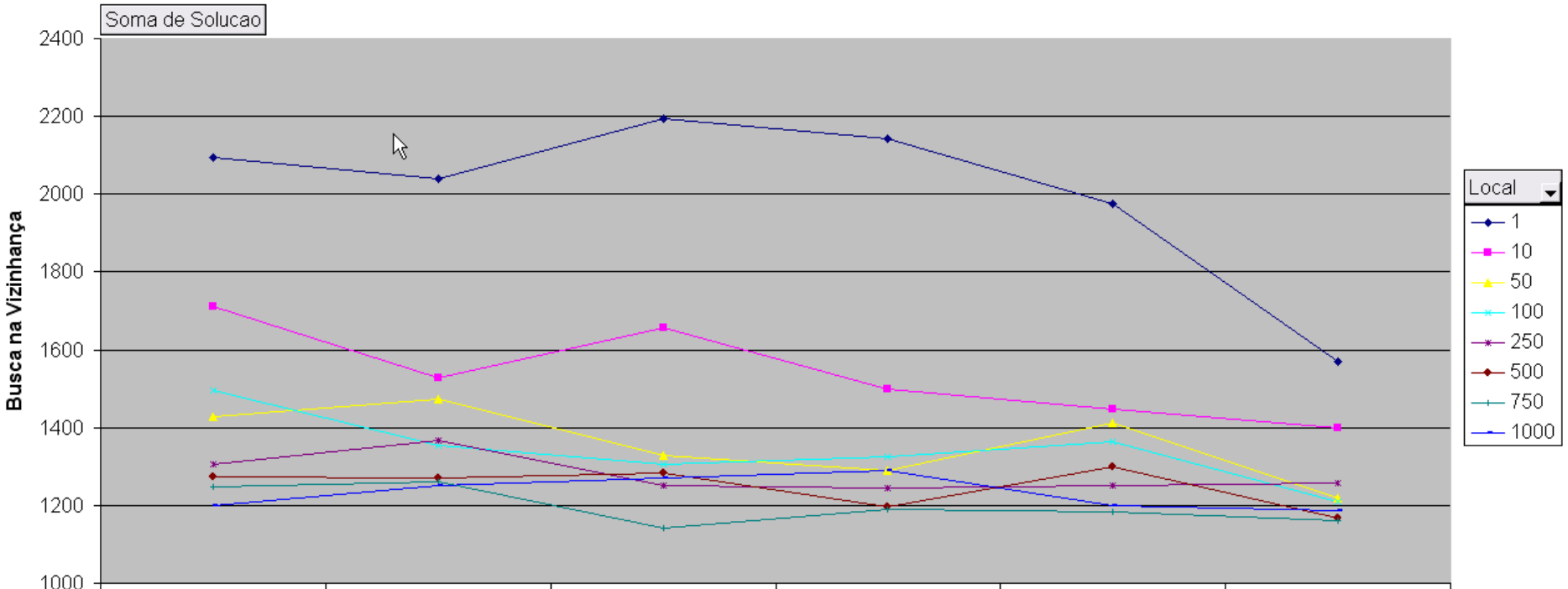



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 932,328 | 948,176 | 931,104 | 921,636 |
| 5000 | 837,534 | 886,498 | 882,046 | 885,891 |
| 10000 | 877,576 | 884,891 | 833,684 | 858,204 |

# Melhor Solução: 833,684

# A-n61-k9 (1)

# Ótimo: 1034



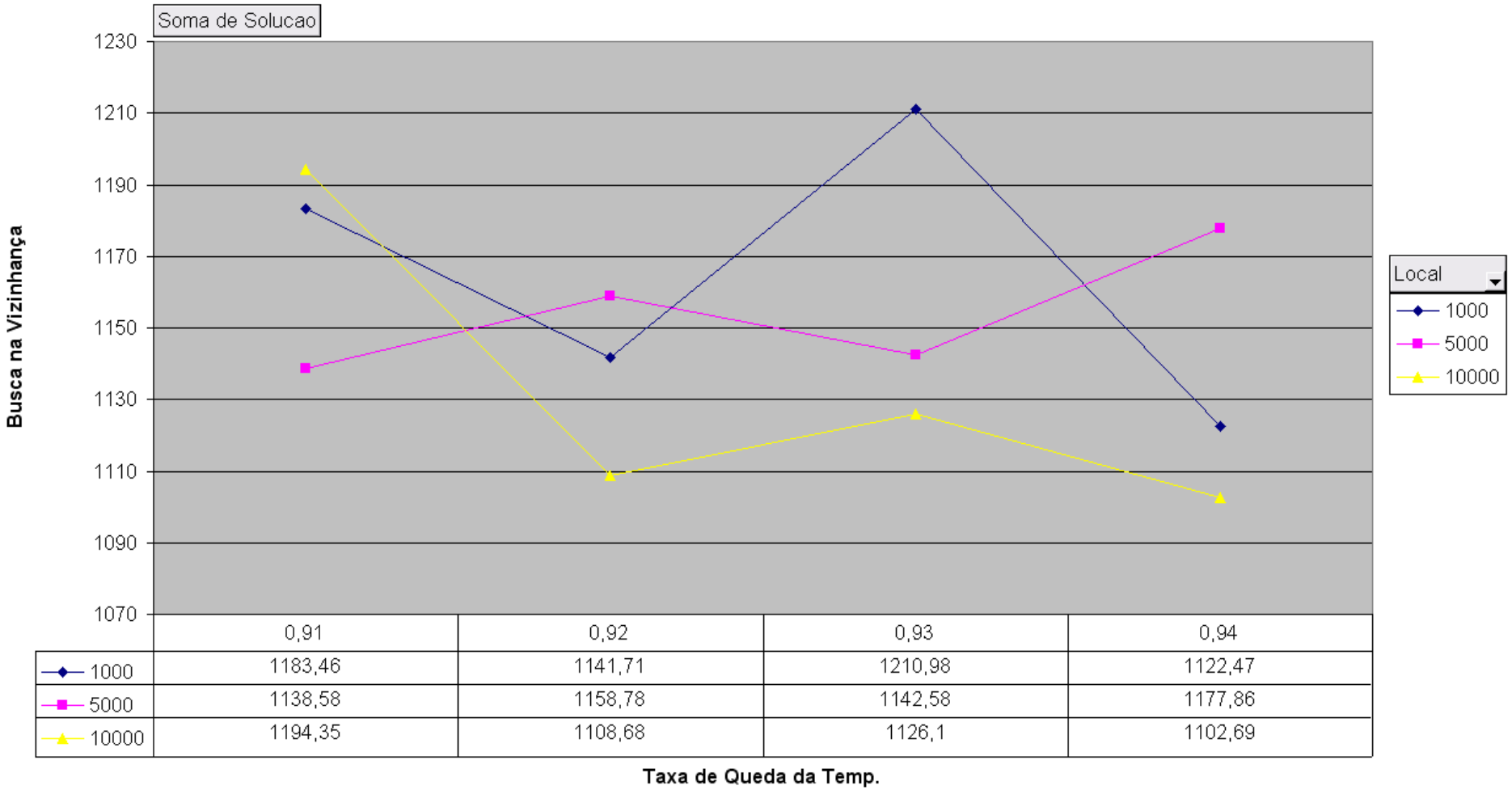

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 2093,51 | 2039,29 | 2193,06 | 2143,34 | 1976,49 | 1568,7 |
| 10 | 1711 | 1526,57 | 1655,57 | 1497,37 | 1446,53 | 1398,96 |
| 50 | 1428,62 | 1473,99 | 1328,32 | 1288,99 | 1412,36 | 1220,14 |
| 100 | 1495,88 | 1355,13 | 1305,47 | 1325,51 | 1364,37 | 1209,6 |
| 250 | 1305,88 | 1367,47 | 1252,3 | 1244,18 | 1251,33 | 1257,77 |
| 500 | 1272,48 | 1269,23 | 1284,53 | 1197,49 | 1299,43 | 1166,36 |
| 750 | 1247,77 | 1259,46 | 1143,03 | 1189,81 | 1183,55 | 1161,3 |
| 1000 | 1199,24 | 1252,09 | 1269,27 | 1290,61 | 1200,61 | 1186,99 |

Taxa de Queda da Temp.

# Melhor Solução: 1143,03

# A-n61-k9 (2)

# Ótimo: 1034

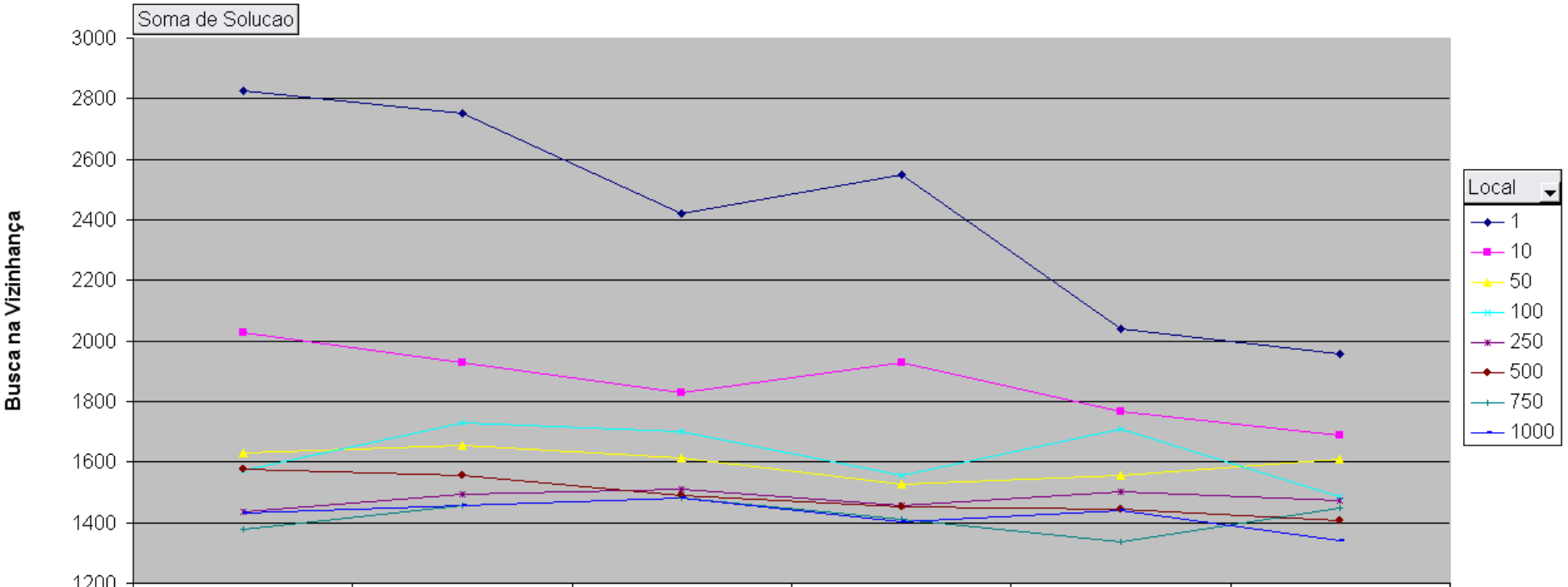



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 1183,46 | 1141,71 | 1210,98 | 1122,47 |
| 5000 | 1138,58 | 1158,78 | 1142,58 | 1177,86 |
| 10000 | 1194,35 | 1108,68 | 1126,1 | 1102,69 |

# Melhor Solução: 1102,69

# A-n65-k9 (1)

# Ótimo: 1174



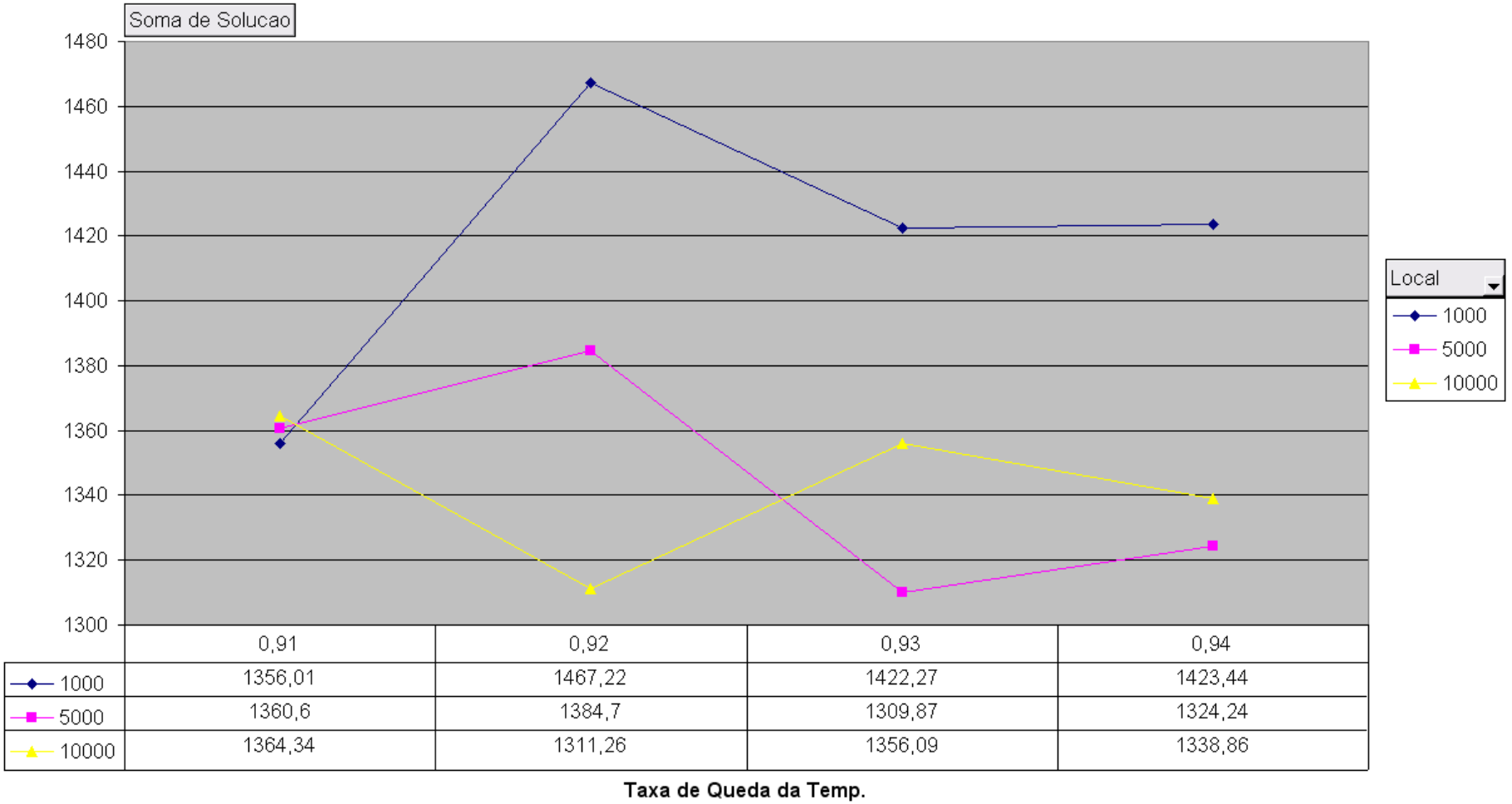

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 2827,16 | 2750,41 | 2421,72 | 2550,27 | 2039,1 | 1957,29 |
| 10 | 2028,3 | 1929,82 | 1827,96 | 1929,35 | 1765,21 | 1689,04 |
| 50 | 1629,33 | 1654,31 | 1612,52 | 1525,68 | 1554,77 | 1608,51 |
| 100 | 1574,4 | 1729,1 | 1699,72 | 1555,71 | 1710,43 | 1484,94 |
| 250 | 1435,06 | 1495,22 | 1509,5 | 1456,28 | 1501,52 | 1472,38 |
| 500 | 1577,57 | 1557,36 | 1487,61 | 1450,61 | 1444,63 | 1407 |
| 750 | 1376,45 | 1456,42 | 1482,14 | 1411,04 | 1337,1 | 1447,47 |
| 1000 | 1433,26 | 1456,48 | 1482,25 | 1403,07 | 1439,23 | 1340,53 |

Taxa de Queda da Temp.

# Melhor Solução: 1337,1

# A-n65-k9 (2)

# Ótimo: 1174

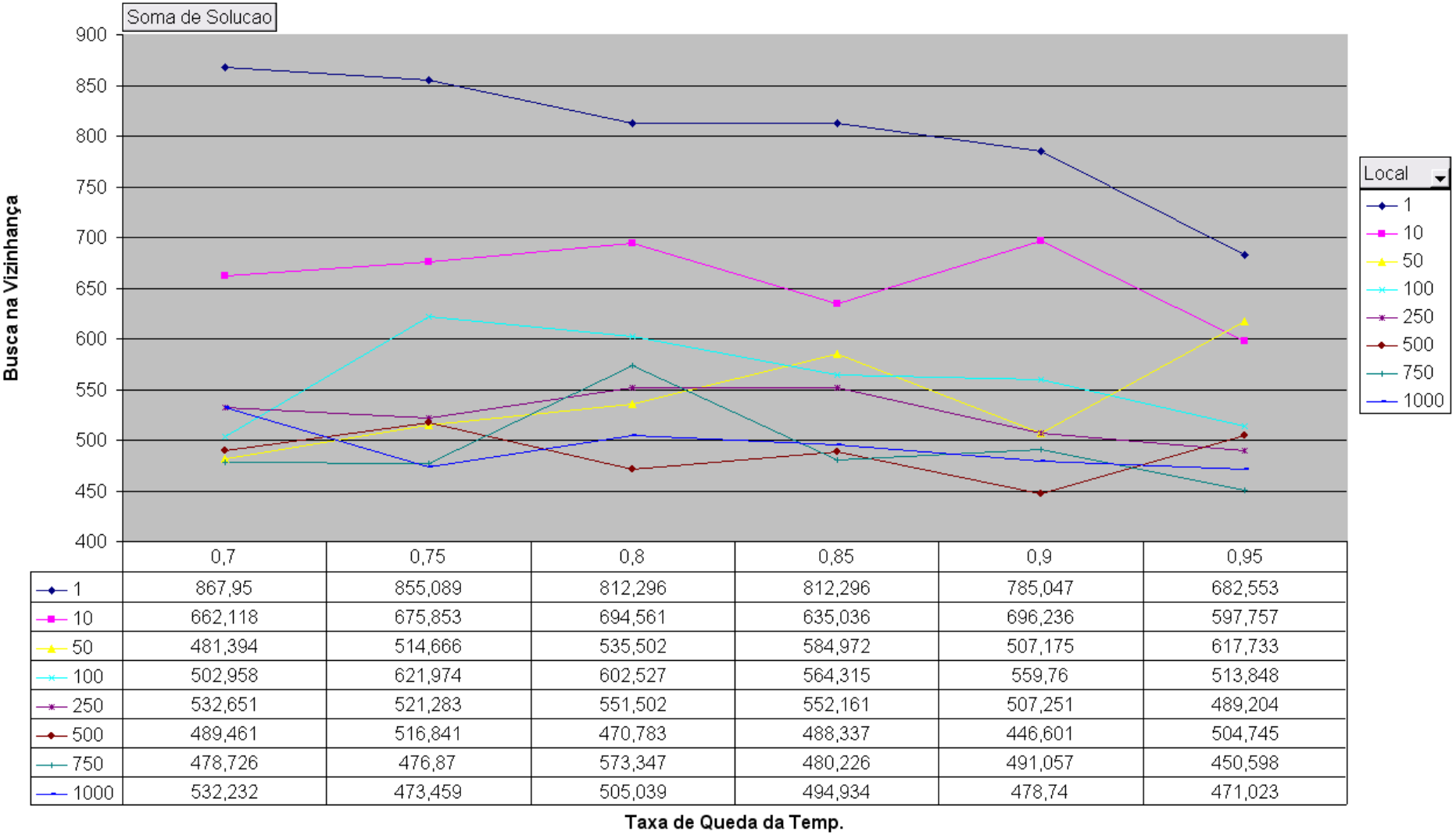



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 1356,01 | 1467,22 | 1422,27 | 1423,44 |
| 5000 | 1360,6 | 1384,7 | 1309,87 | 1324,24 |
| 10000 | 1364,34 | 1311,26 | 1356,09 | 1338,86 |

# Melhor Solução: 1309,87

# E-n30-k3 (1)

# Ótimo: 534

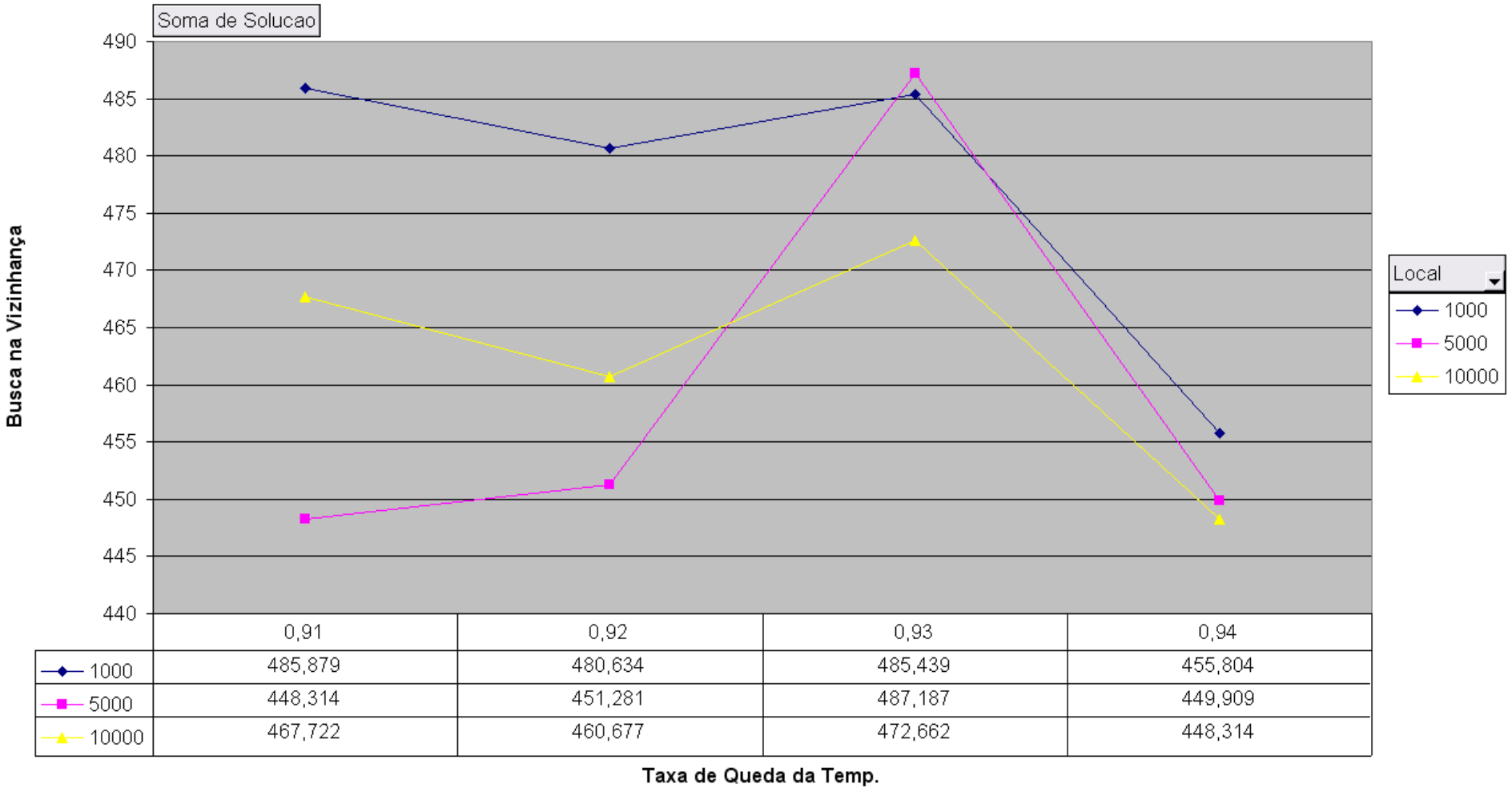



| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 867,95 | 855,089 | 812,296 | 812,296 | 785,047 | 682,553 |
| 10 | 662,118 | 675,853 | 694,561 | 635,036 | 696,236 | 597,757 |
| 50 | 481,394 | 514,666 | 535,502 | 584,972 | 507,175 | 617,733 |
| 100 | 502,958 | 621,974 | 602,527 | 564,315 | 559,76 | 513,848 |
| 250 | 532,651 | 521,283 | 551,502 | 552,161 | 507,251 | 489,204 |
| 500 | 489,461 | 516,841 | 470,783 | 488,337 | 446,601 | 504,745 |
| 750 | 478,726 | 476,87 | 573,347 | 480,226 | 491,057 | 450,598 |
| 1000 | 532,232 | 473,459 | 505,039 | 494,934 | 478,74 | 471,023 |

# Melhor Solução: 446,601 (?)

# E-n30-k3 (2)

# Ótimo: 534

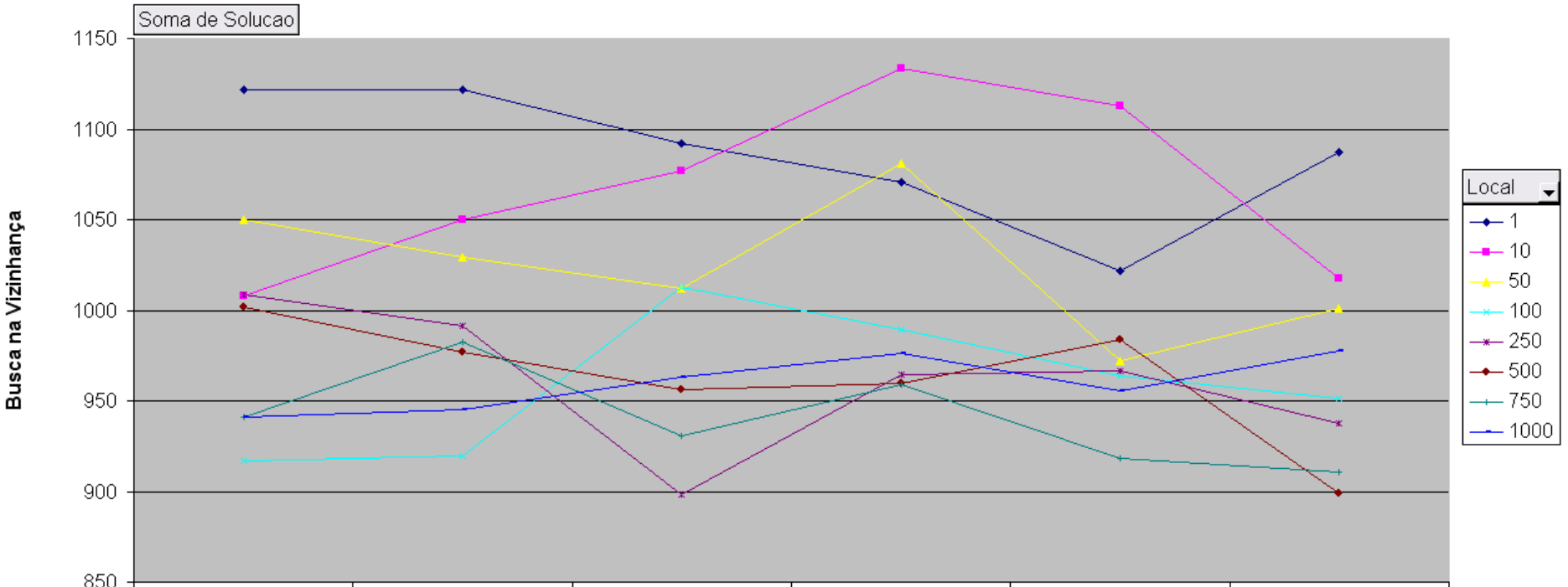



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 485,879 | 480,634 | 485,439 | 455,804 |
| 5000 | 448,314 | 451,281 | 487,187 | 449,909 |
| 10000 | 467,722 | 460,677 | 472,662 | 448,314 |

# Melhor Solução: 448,314 (?)

# E-n33-k4 (1)

# Ótimo: 835



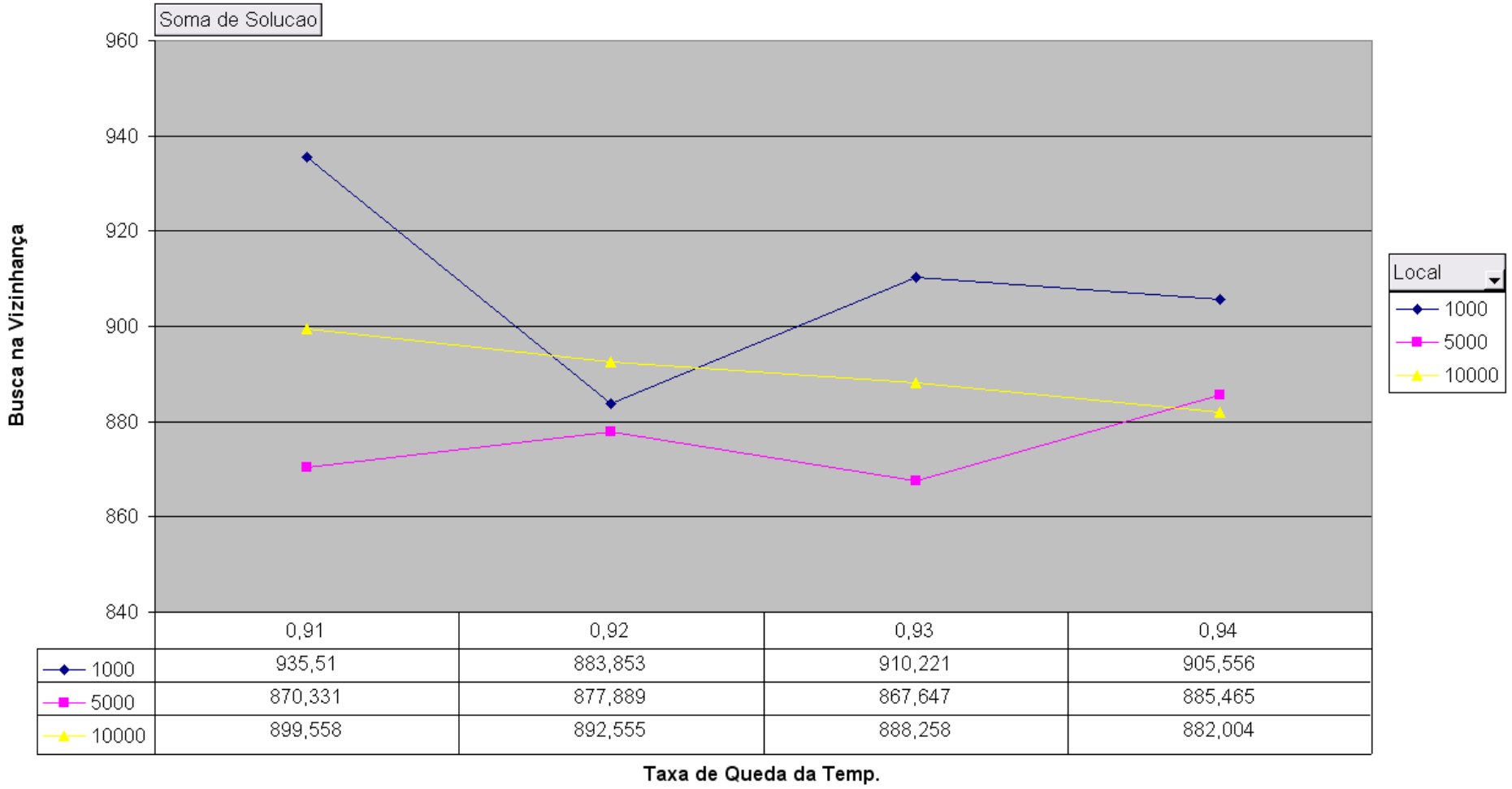

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1121,53 | 1121,81 | 1092,19 | 1070,62 | 1021,47 | 1087,52 |
| 10 | 1008,06 | 1050,02 | 1076,75 | 1133,73 | 1112,79 | 1017,46 |
| 50 | 1050,13 | 1029,55 | 1011,89 | 1080,93 | 972,053 | 1000,93 |
| 100 | 916,553 | 919,765 | 1012,93 | 989,097 | 963,454 | 951,1 |
| 250 | 1008,35 | 991,36 | 898,235 | 964,515 | 966,226 | 937,913 |
| 500 | 1002 | 976,738 | 956,231 | 959,997 | 983,566 | 898,636 |
| 750 | 940,751 | 982,542 | 930,985 | 959,134 | 918,484 | 910,375 |
| 1000 | 941,018 | 945,396 | 962,982 | 976,281 | 955,762 | 977,808 |

Taxa de Queda da Temp.

# Melhor Solução: 898,235

# E-n33-k4 (2)

# Ótimo: 835

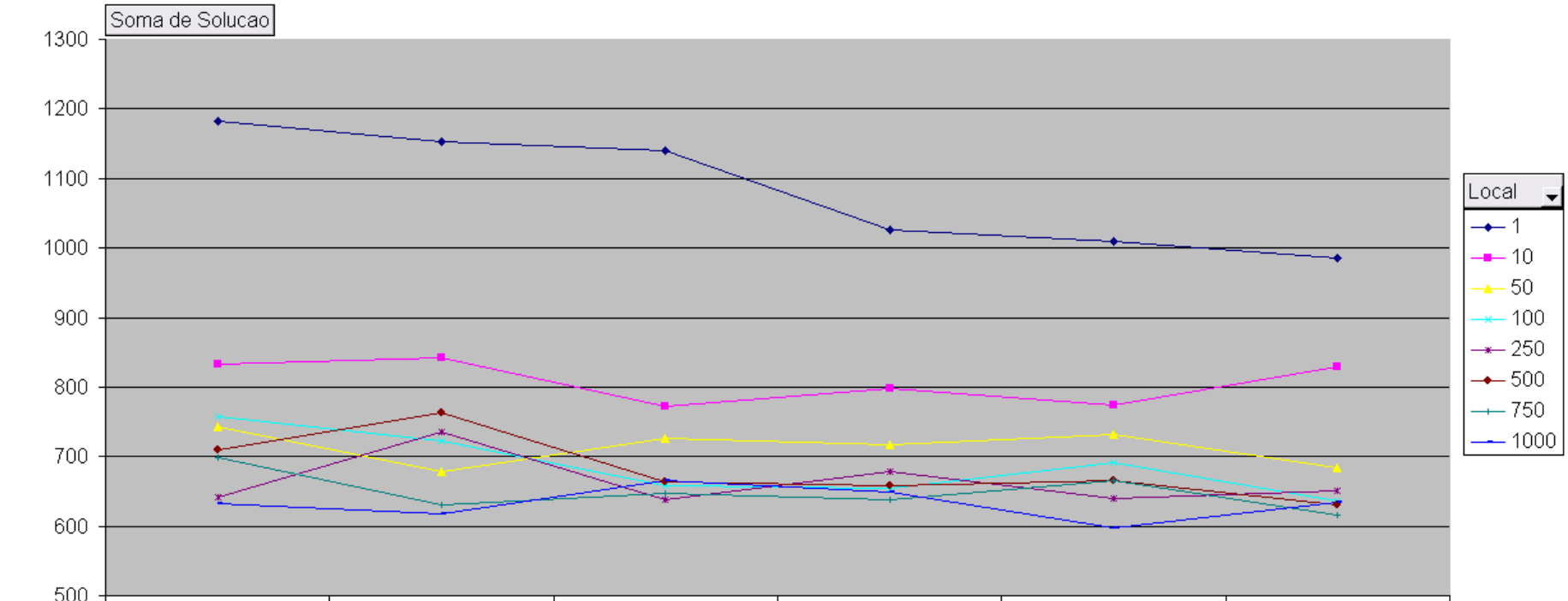



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 935,51 | 883,853 | 910,221 | 905,556 |
| 5000 | 870,331 | 877,889 | 867,647 | 885,465 |
| 10000 | 899,558 | 892,555 | 888,258 | 882,004 |

# Melhor Solução: 867,647

# E-n51-k5 (1)

# Ótimo: 521



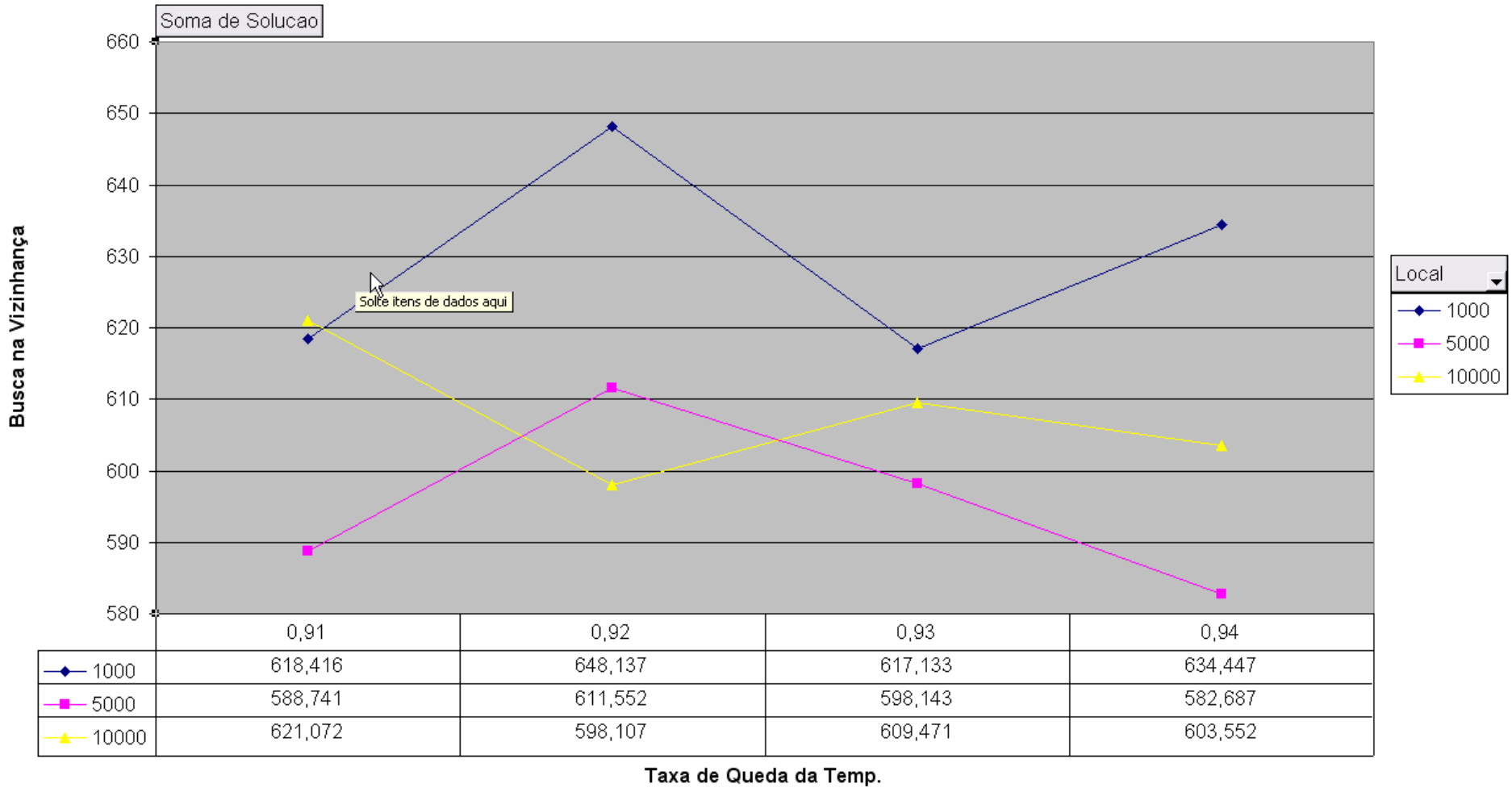

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1183 | 1153,46 | 1139,88 | 1025,61 | 1009,12 | 985,071 |
| 10 | 833,168 | 842,356 | 772,607 | 797,218 | 774,4 | 829,431 |
| 50 | 742,749 | 678,99 | 726,764 | 717,928 | 731,981 | 684,244 |
| 100 | 757,757 | 722,779 | 657,455 | 654,763 | 690,621 | 636,435 |
| 250 | 640,795 | 735,91 | 638,453 | 679,248 | 639,957 | 650,827 |
| 500 | 709,809 | 763,075 | 663,578 | 658,531 | 665,532 | 629,849 |
| 750 | 698,101 | 630,487 | 647,227 | 637,319 | 665,77 | 615,326 |
| 1000 | 632,068 | 617,195 | 665,67 | 648,262 | 598,002 | 634,692 |

Taxa de Queda da Temp.

# Melhor Solução: 598,002

# E-n51-k5 (2)

# Ótimo: 521

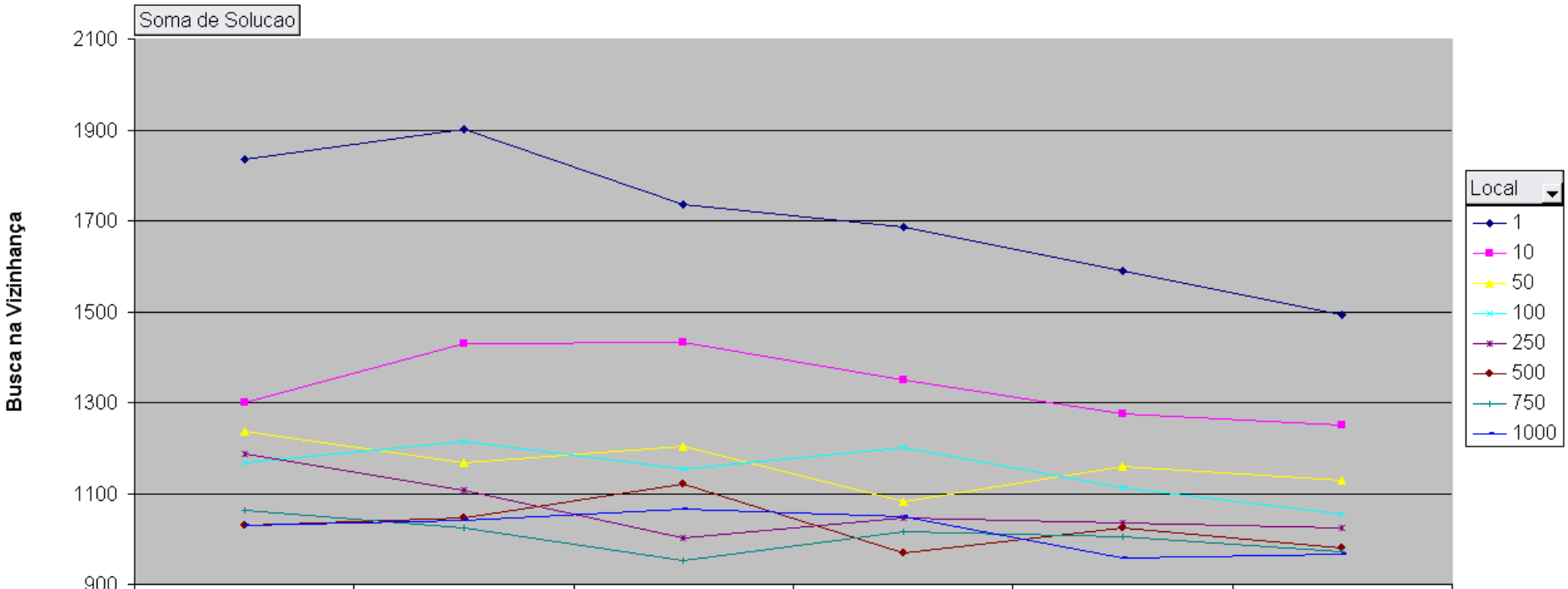



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 618,416 | 648,137 | 617,133 | 634,447 |
| 5000 | 588,741 | 611,552 | 598,143 | 582,687 |
| 10000 | 621,072 | 598,107 | 609,471 | 603,552 |

# Melhor Solução: 582,587

# E-n76-k10 (1)

## Ótimo: 830



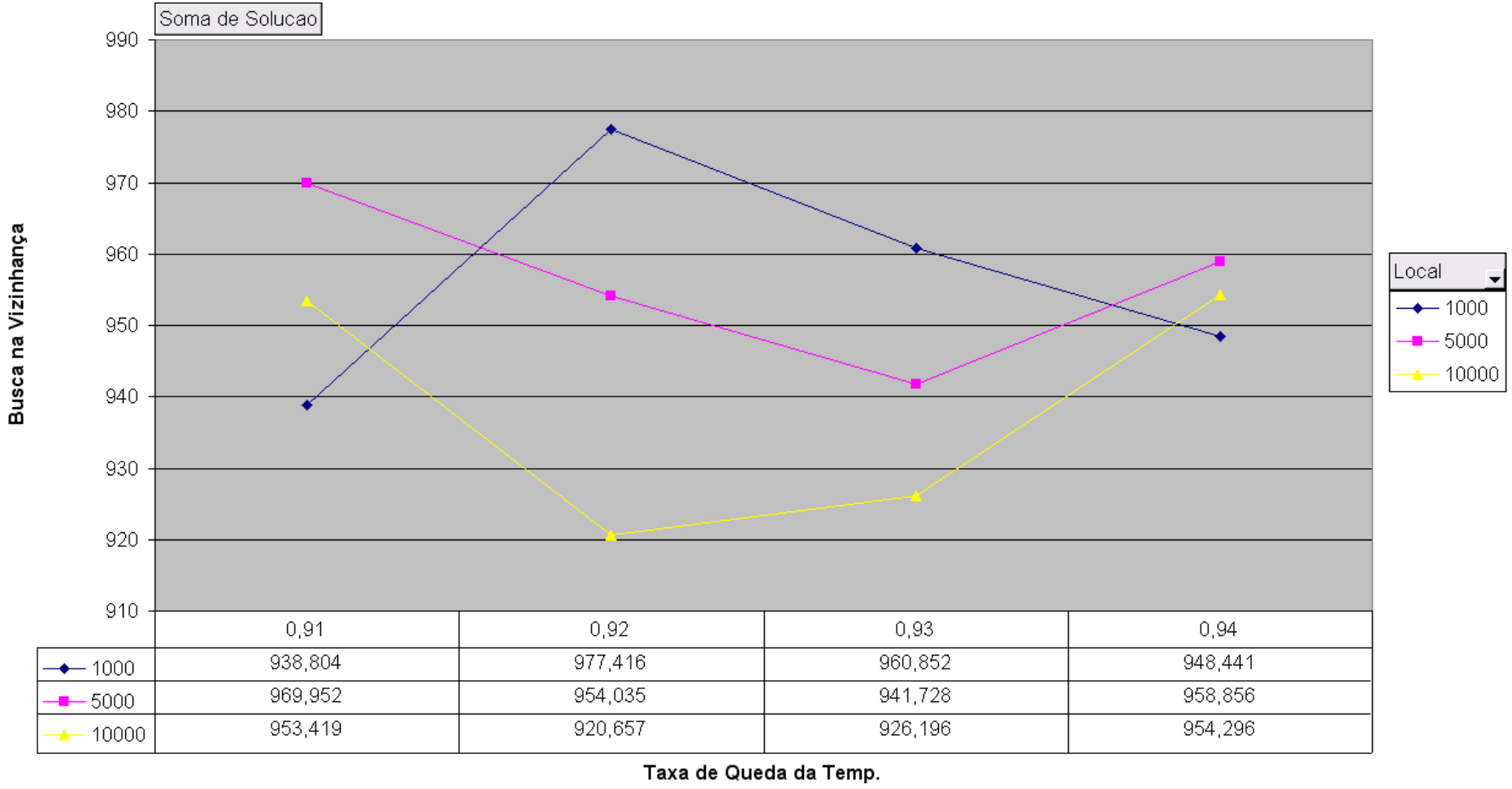

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1836,33 | 1901,17 | 1735,42 | 1686,86 | 1590,75 | 1492,35 |
| 10 | 1299,48 | 1431,03 | 1432,49 | 1348,92 | 1276,46 | 1249,39 |
| 50 | 1236,19 | 1167,73 | 1202,92 | 1082,17 | 1160,65 | 1128,18 |
| 100 | 1166,73 | 1214,42 | 1154,3 | 1201,19 | 1111,23 | 1054,11 |
| 250 | 1186,31 | 1106,61 | 1000,99 | 1046,51 | 1035,21 | 1025,04 |
| 500 | 1028,8 | 1047,49 | 1121,53 | 967,642 | 1023,86 | 979,354 |
| 750 | 1062,12 | 1023,25 | 951,55 | 1015,87 | 1005,34 | 970,481 |
| 1000 | 1028,29 | 1040,43 | 1066,46 | 1049,58 | 958,617 | 965,186 |

Taxa de Queda da Temp.

## Melhor Solução: 958,617

# E-n76-k10 (2)

Ótimo: 830

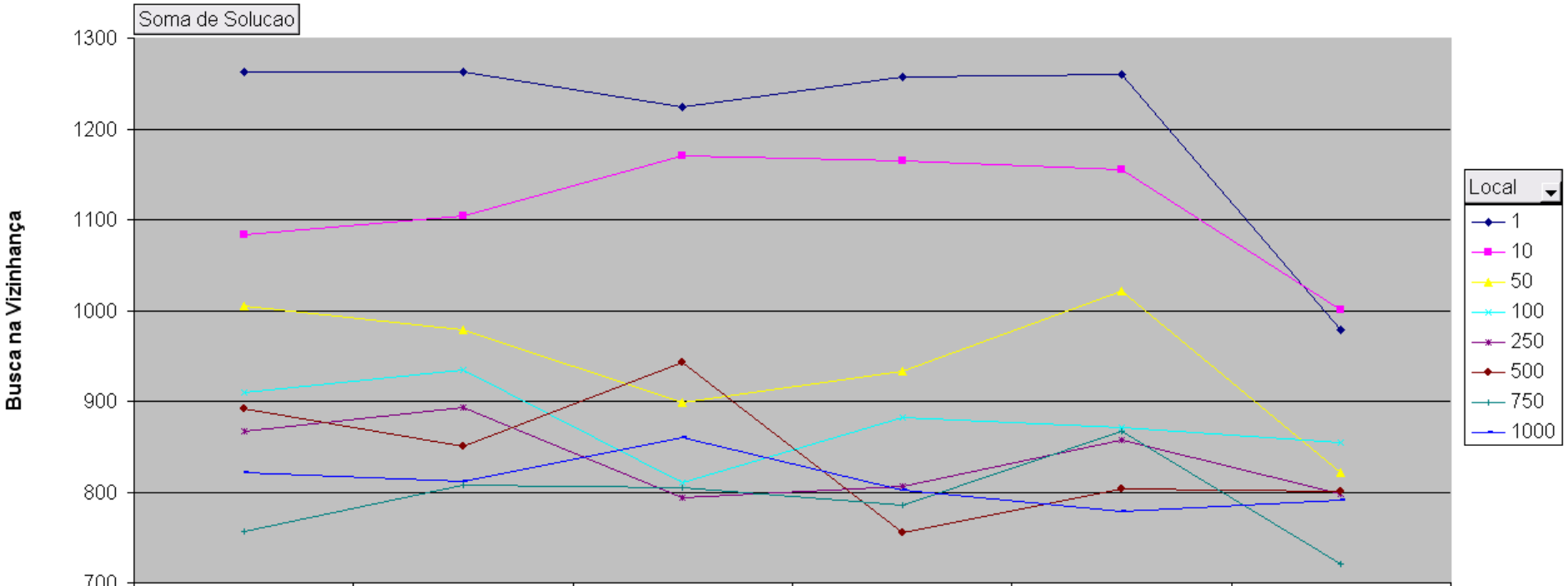



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 938,804 | 977,416 | 960,852 | 948,441 |
| 5000 | 969,952 | 954,035 | 941,728 | 958,856 |
| 10000 | 953,419 | 920,657 | 926,196 | 954,296 |

Melhor Solução: 920,657

# F-n45-k4 (1)

# Ótimo: 724



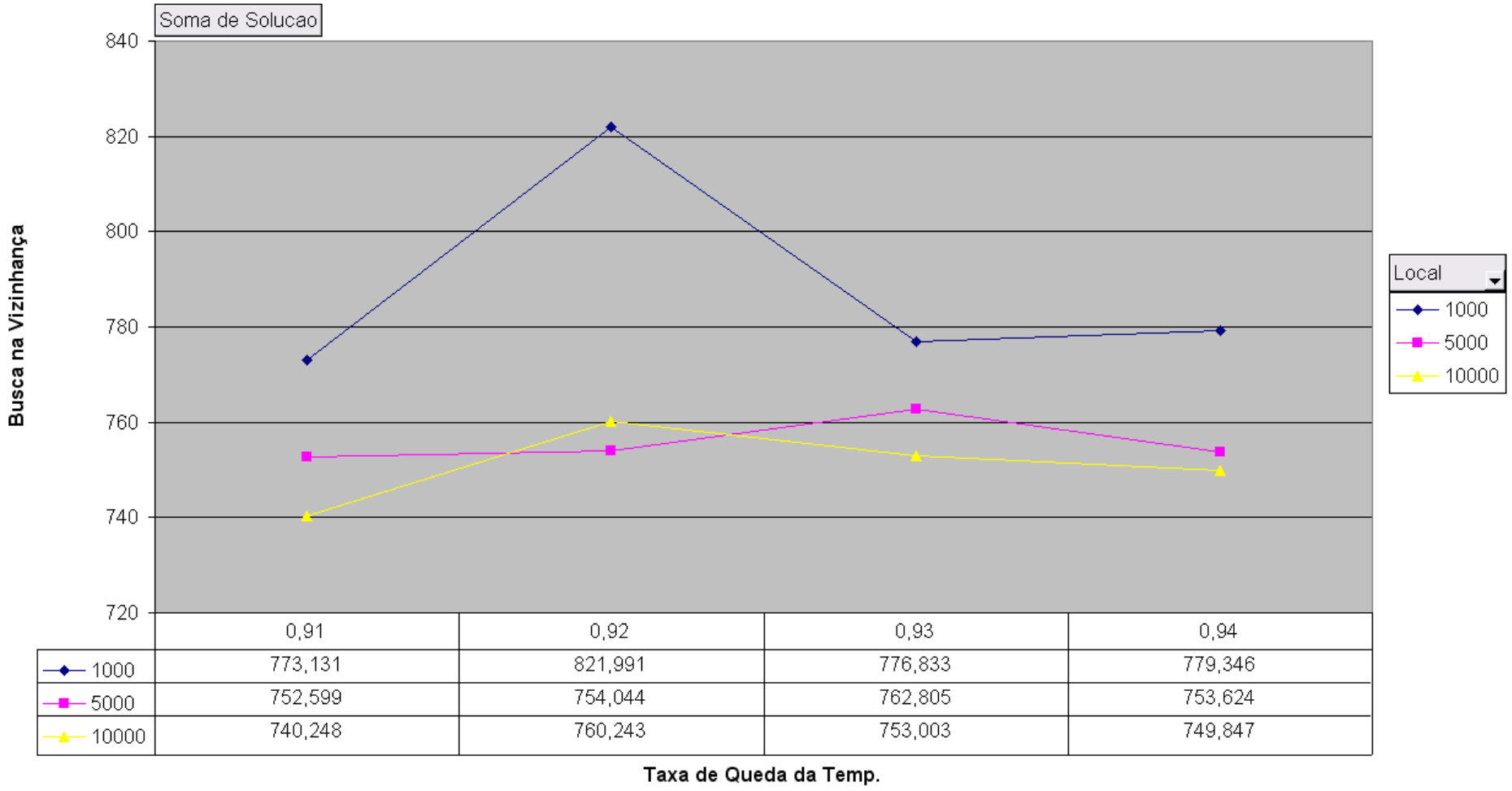

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 1262,92 | 1262,92 | 1224,4 | 1257,82 | 1260,36 | 979,144 |
| 10 | 1083,58 | 1104,06 | 1170,39 | 1164,34 | 1155,74 | 1001,33 |
| 50 | 1004,7 | 978,708 | 898,185 | 933,009 | 1021,09 | 821,454 |
| 100 | 909,395 | 933,813 | 810,306 | 882,576 | 870,665 | 854,231 |
| 250 | 867,185 | 892,966 | 793,637 | 806,179 | 857,256 | 798,62 |
| 500 | 892,193 | 850,136 | 942,937 | 755,479 | 803,56 | 800,157 |
| 750 | 756,028 | 806,919 | 805,199 | 786,103 | 866,218 | 721,071 |
| 1000 | 821,873 | 811,906 | 860,364 | 802,143 | 778,336 | 791,587 |

Taxa de Queda da Temp.

# Melhor Solução: 721,071 (?)

# F-n45-k4 (2)

# Ótimo: 724

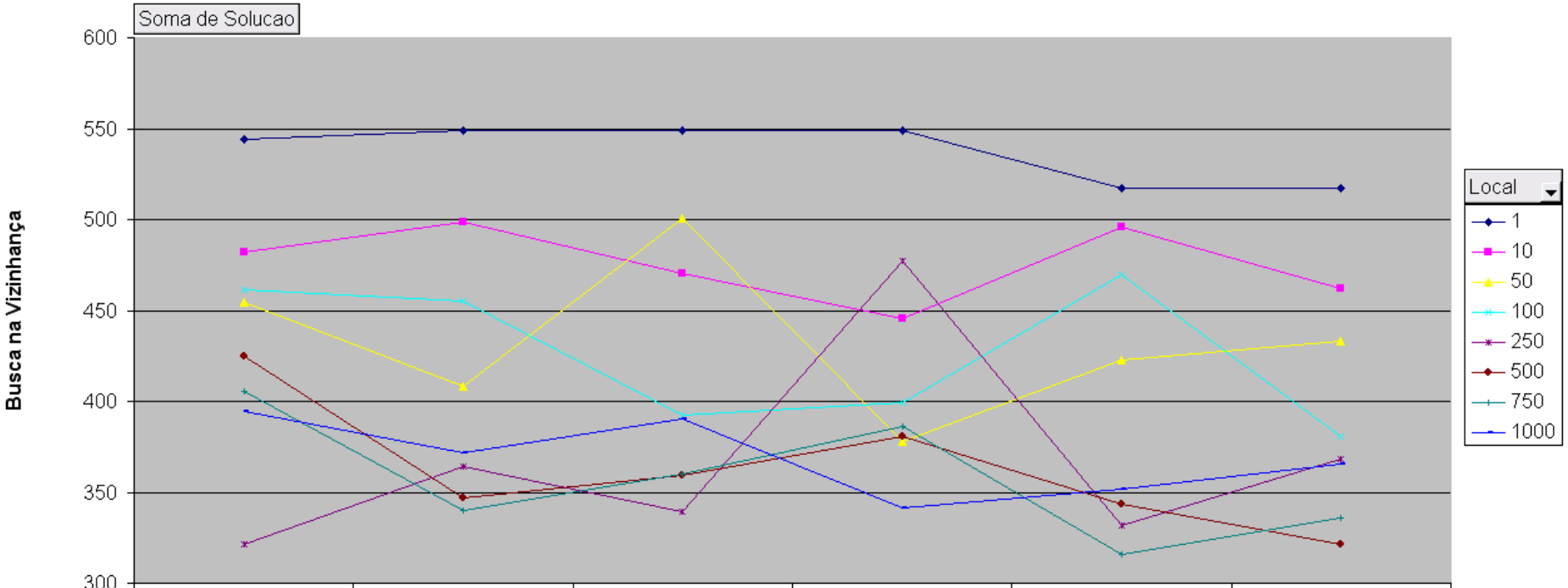



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 773,131 | 821,991 | 776,833 | 779,346 |
| 5000 | 752,599 | 754,044 | 762,805 | 753,624 |
| 10000 | 740,248 | 760,243 | 753,003 | 749,847 |

# Melhor Solução: 740,248

# F-n72-k4 (1)

# Ótimo: 237



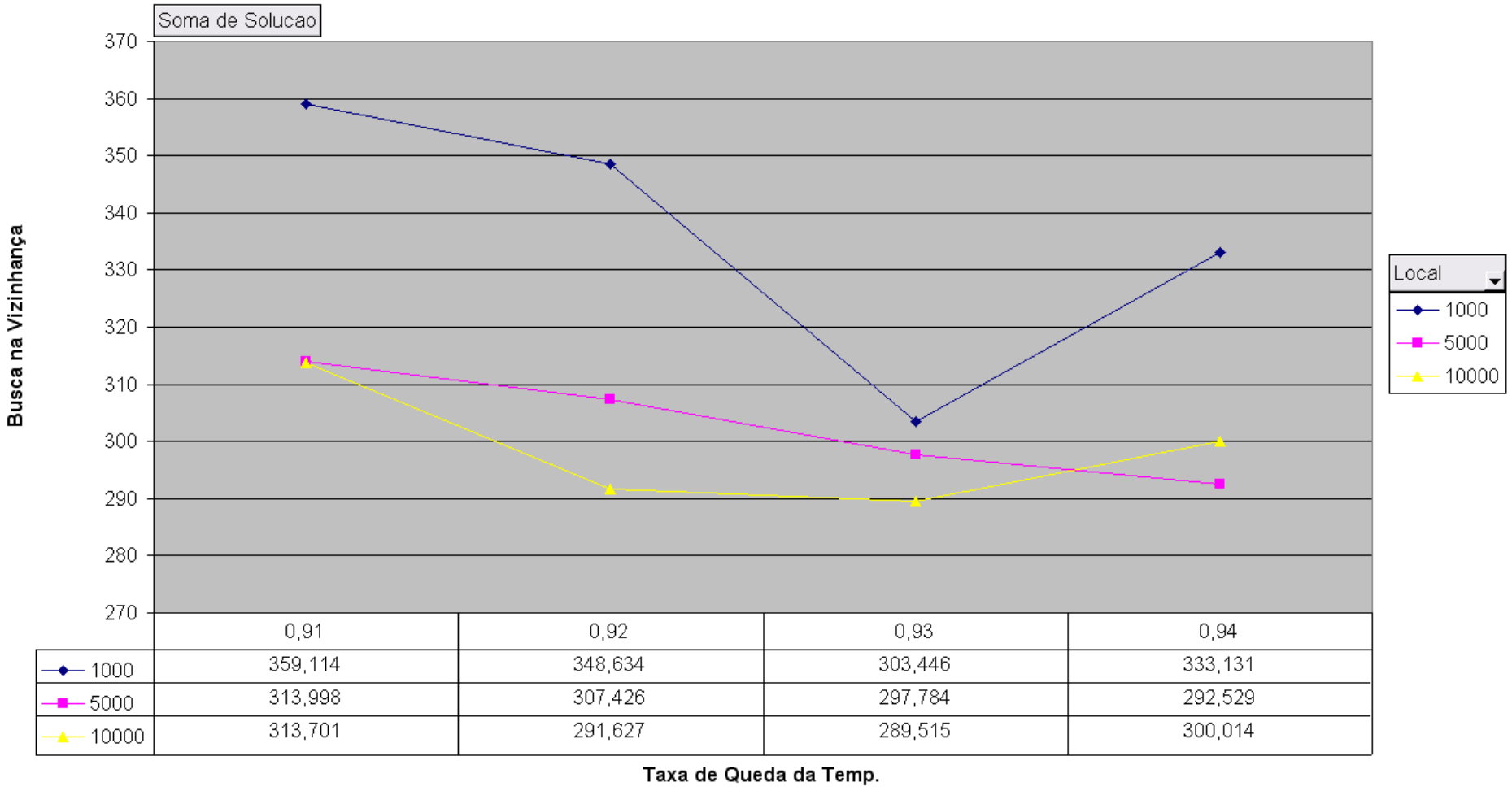

| | 0,7 | 0,75 | 0,8 | 0,85 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|
| 1 | 544,447 | 548,977 | 548,977 | 548,977 | 516,969 | 517,421 |
| 10 | 481,895 | 498,433 | 470,425 | 445,629 | 496,075 | 462,4 |
| 50 | 454,245 | 408,394 | 500,855 | 378,228 | 422,624 | 433,227 |
| 100 | 461,375 | 455,31 | 392,394 | 399,651 | 469,474 | 380,432 |
| 250 | 321,311 | 364,194 | 339,105 | 477,573 | 332,066 | 367,971 |
| 500 | 424,552 | 347,042 | 359,147 | 380,355 | 343,256 | 321,168 |
| 750 | 405,436 | 339,656 | 360,211 | 386 | 315,926 | 335,823 |
| 1000 | 394,546 | 372,004 | 390,093 | 341,458 | 351,945 | 365,853 |

Taxa de Queda da Temp.

# Melhor Solução: 315,926

# F-n72-k4 (2)

# Ótimo: 237

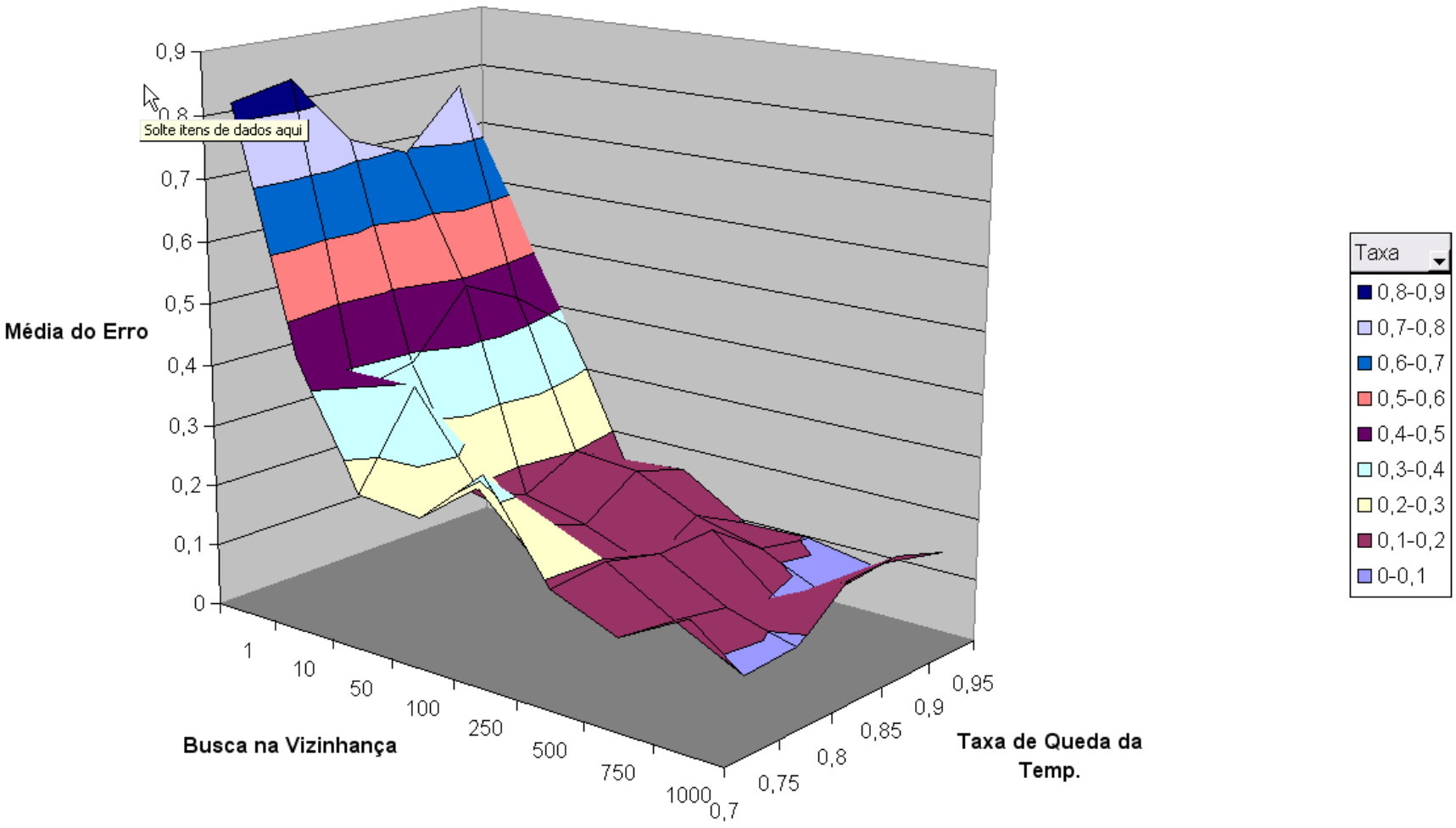



| | 0,91 | 0,92 | 0,93 | 0,94 |
|---|---|---|---|---|
| 1000 | 359,114 | 348,634 | 303,446 | 333,131 |
| 5000 | 313,998 | 307,426 | 297,784 | 292,529 |
| 10000 | 313,701 | 291,627 | 289,515 | 300,014 |

# Melhor Solução: 289,515

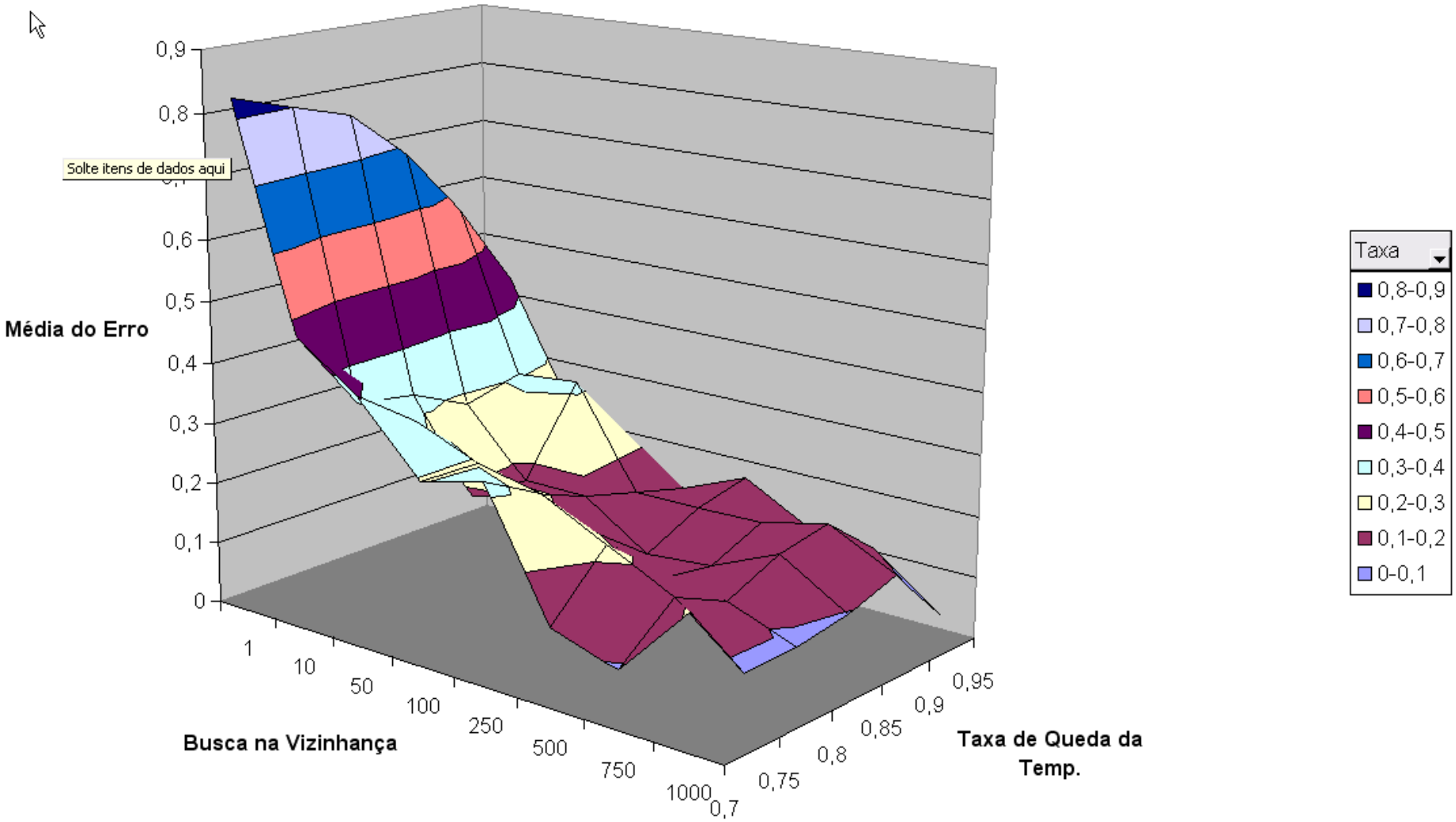
# Parâmetros x Erro (1)

## A-n32-k5



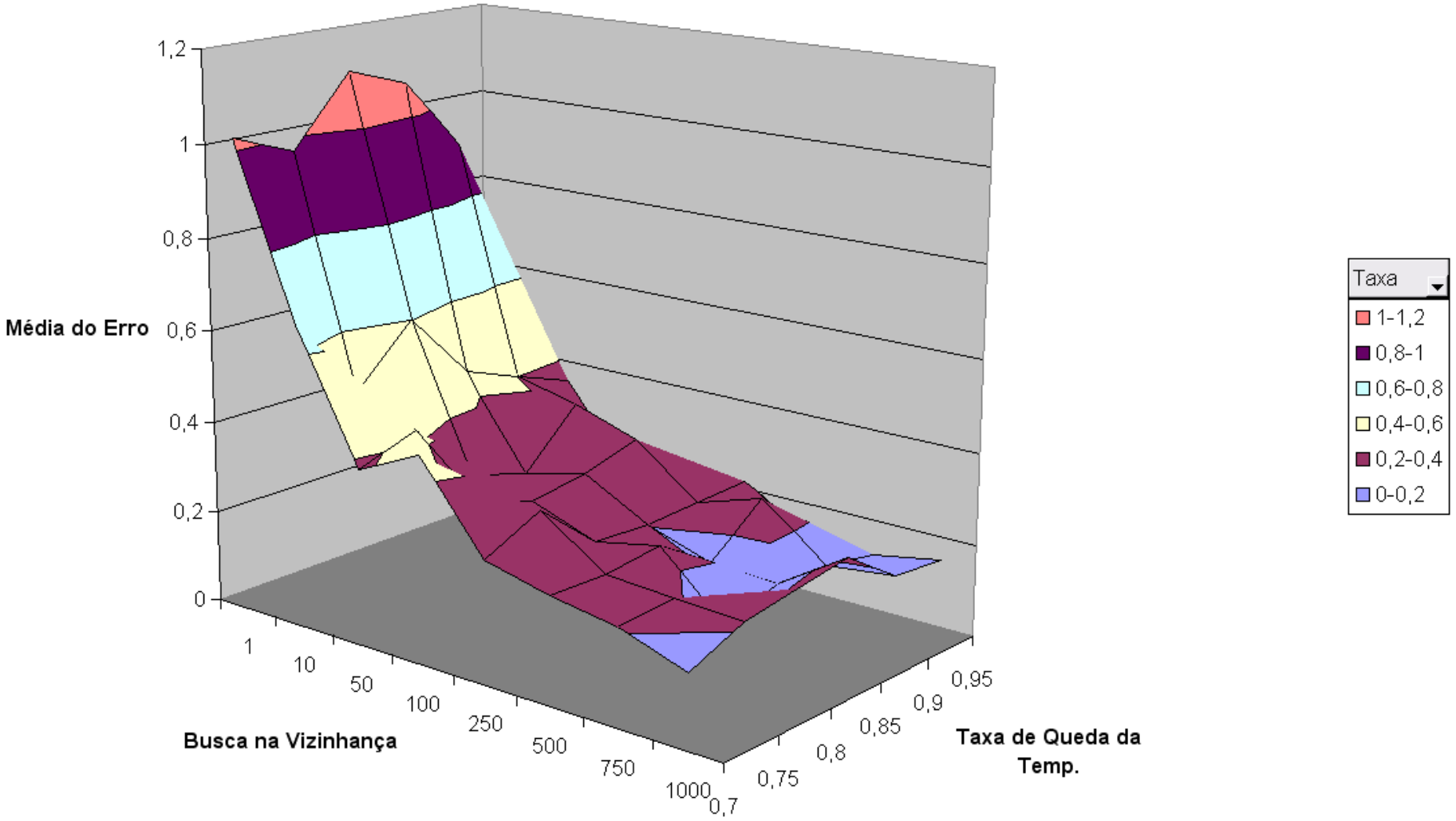
# Parâmetros x Erro (2)

## A-n39-k6



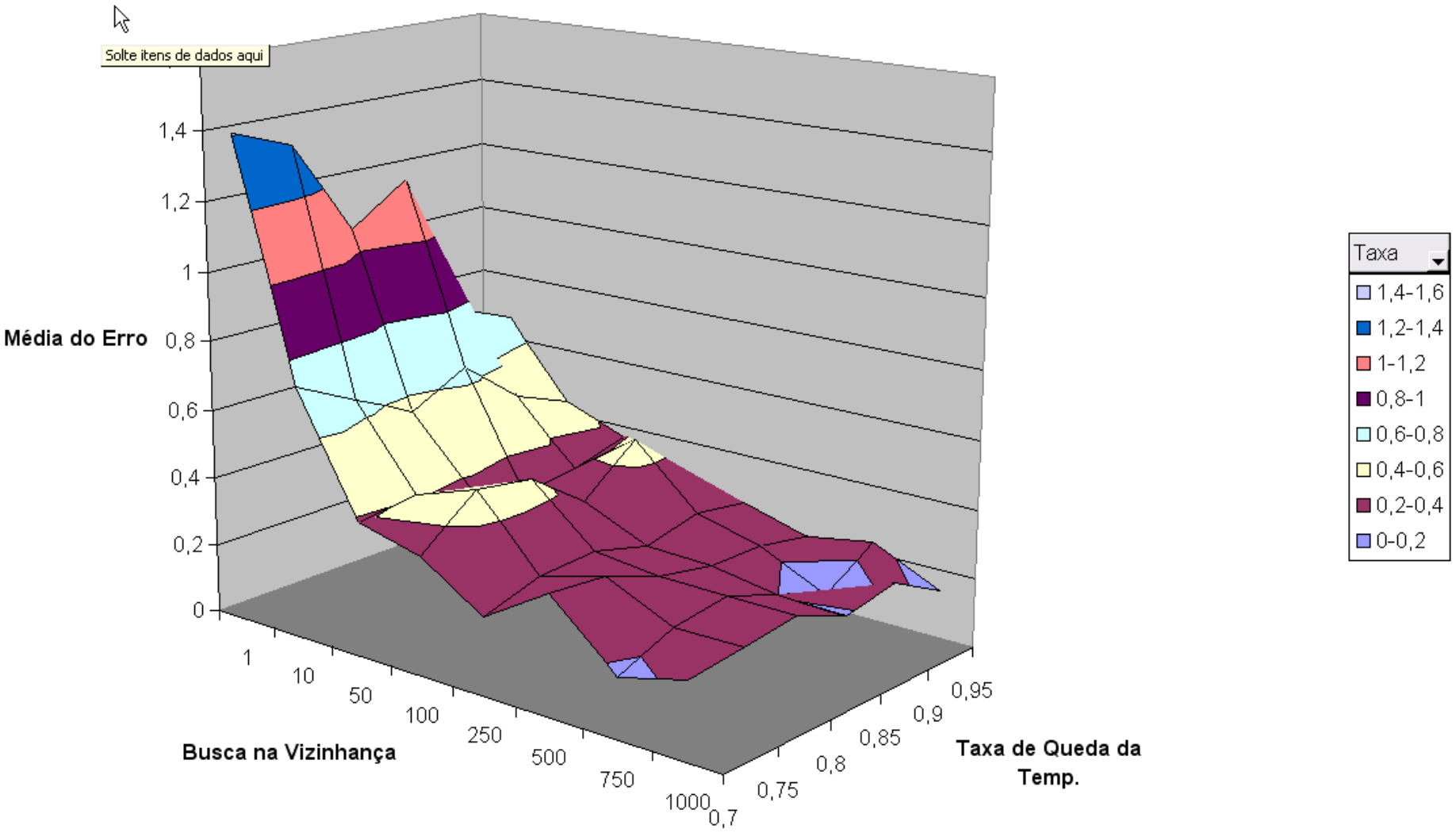
# Parâmetros x Erro (3)

## A-n61-k9



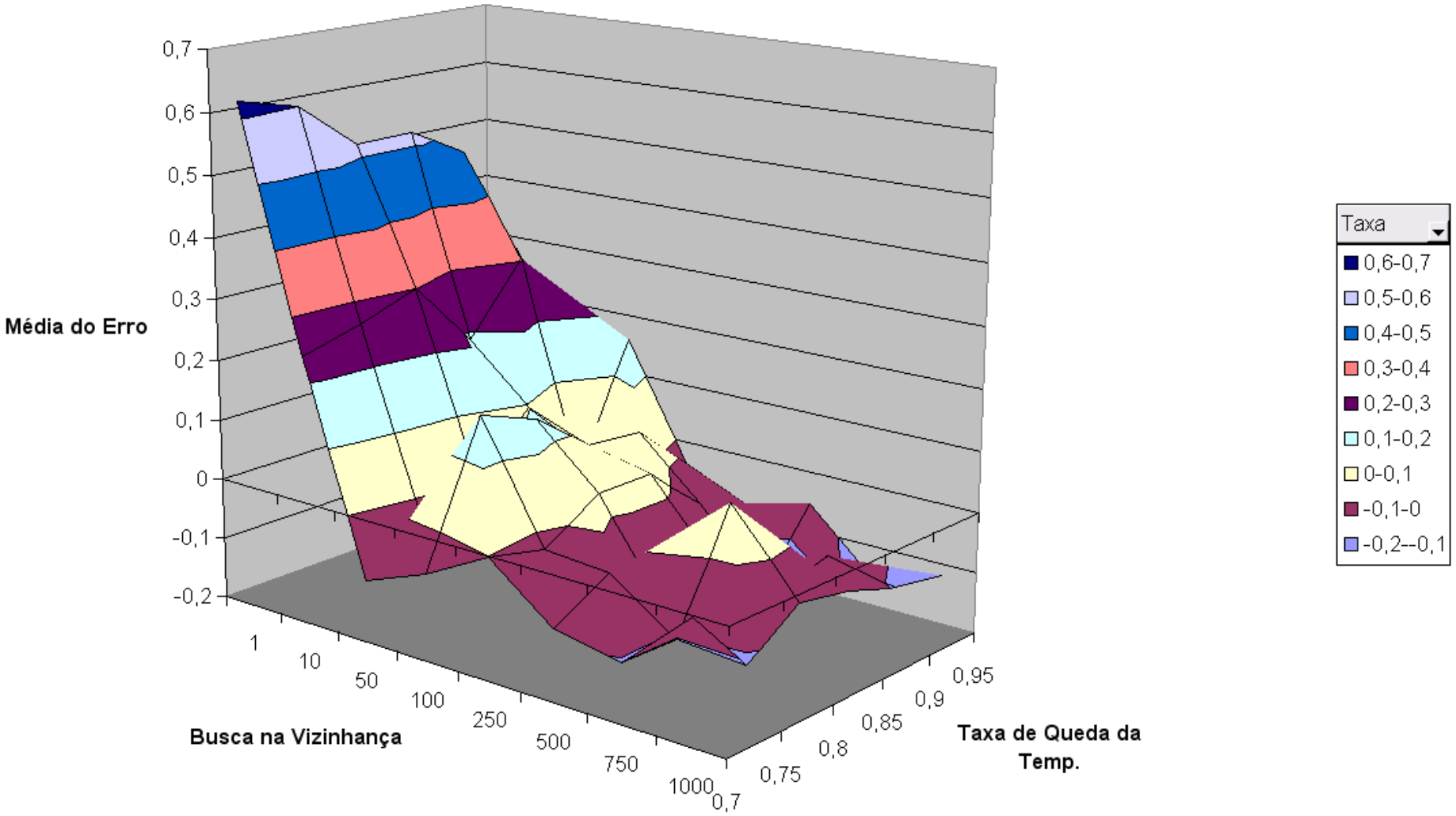
# Parâmetros x Erro (4)

## A-n65-k9



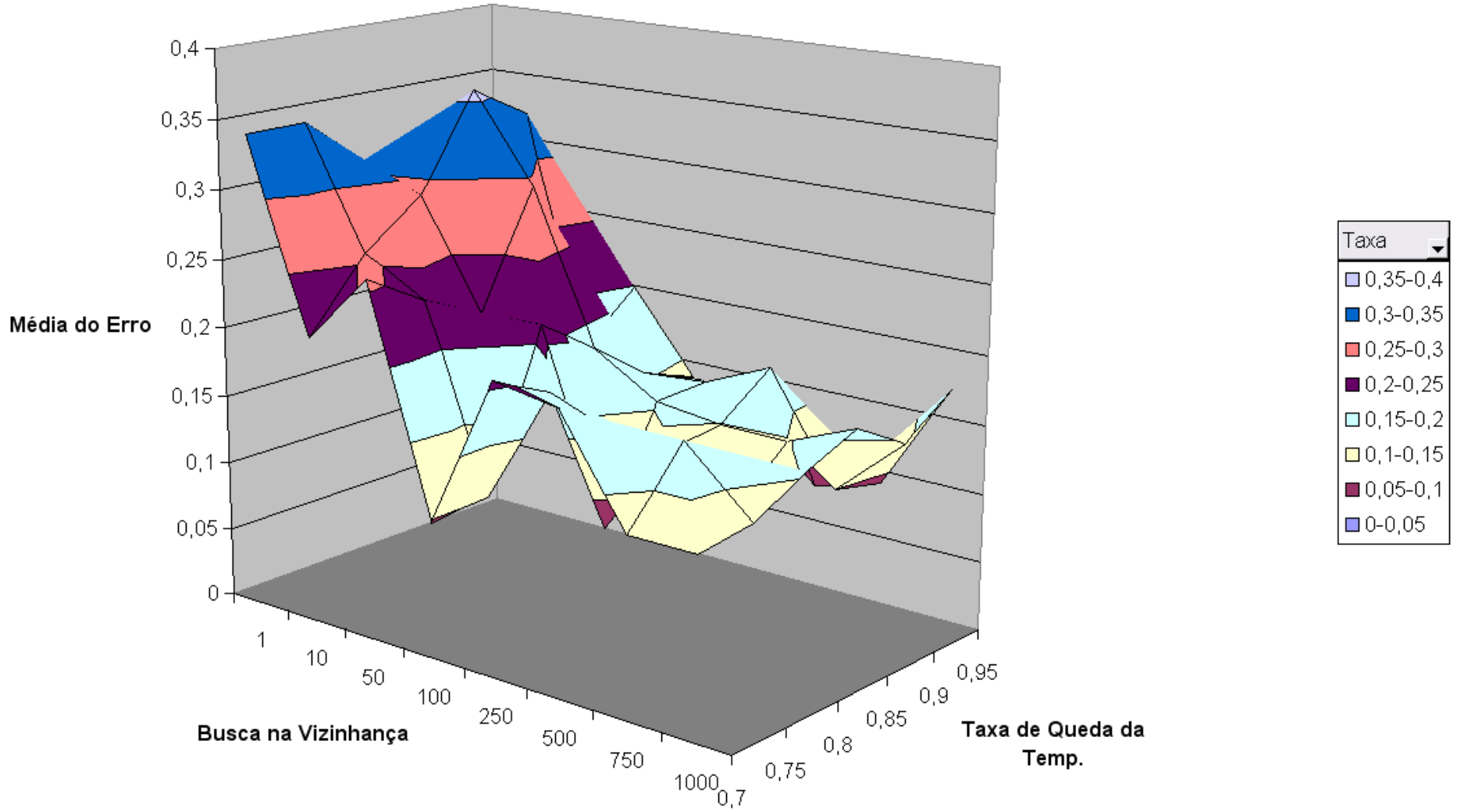
# Parâmetros x Erro (5)

## E-n30-k3



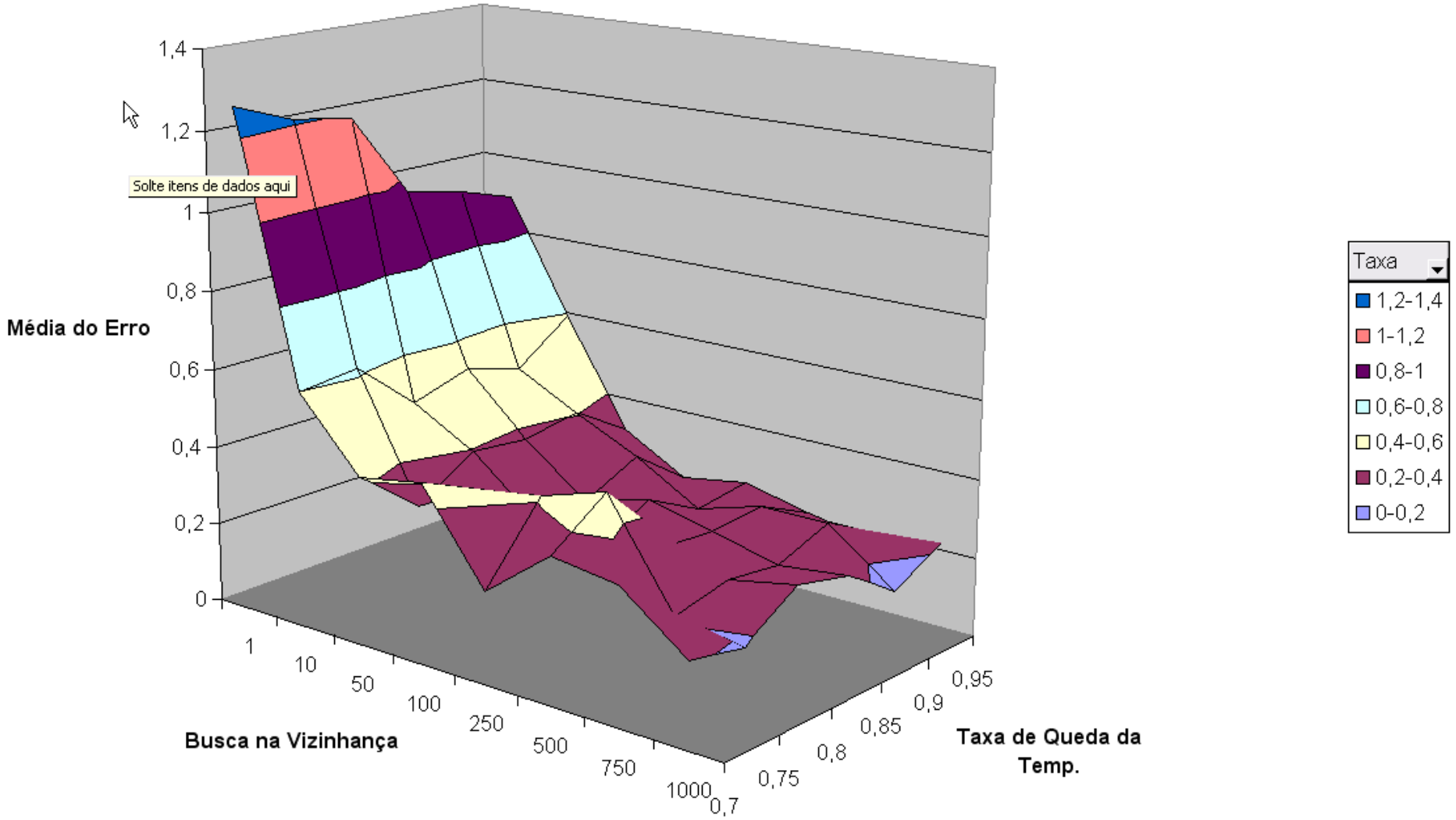
# Parâmetros x Erro (6)

## E-n33-k4



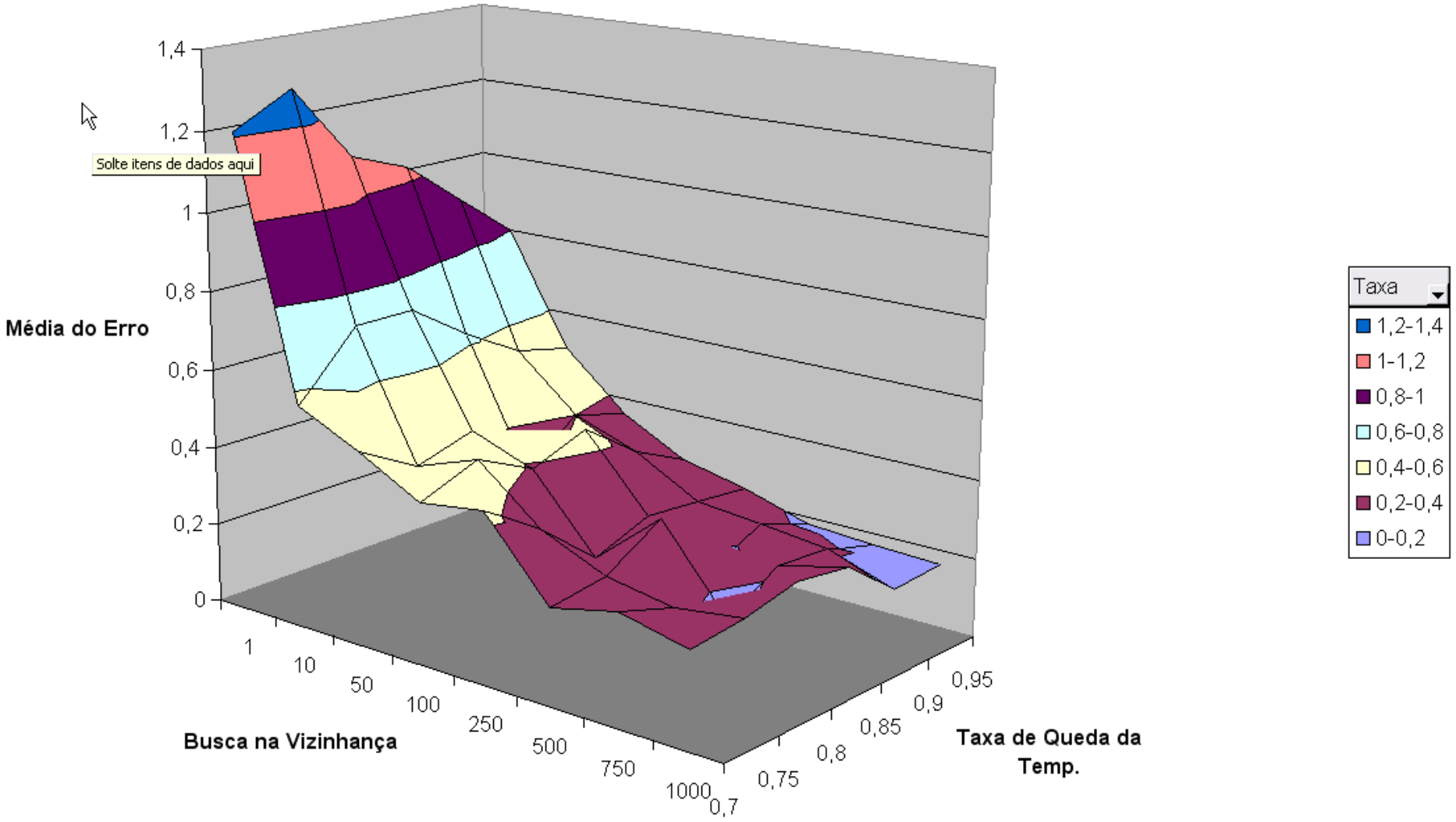
# Parâmetros x Erro (7)

## E-n51-k5



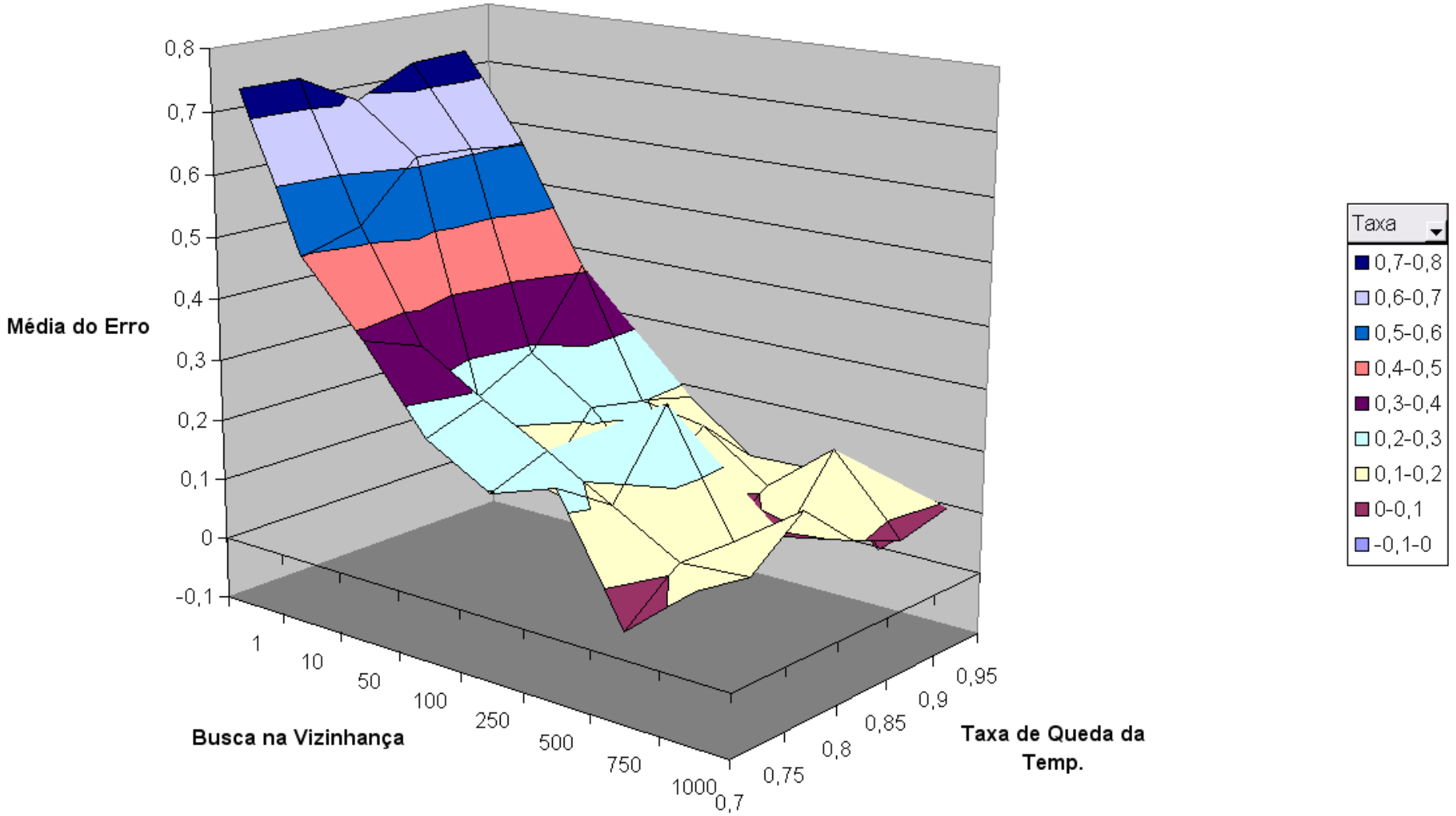
# Parâmetros x Erro (8)

## E-n76-k10



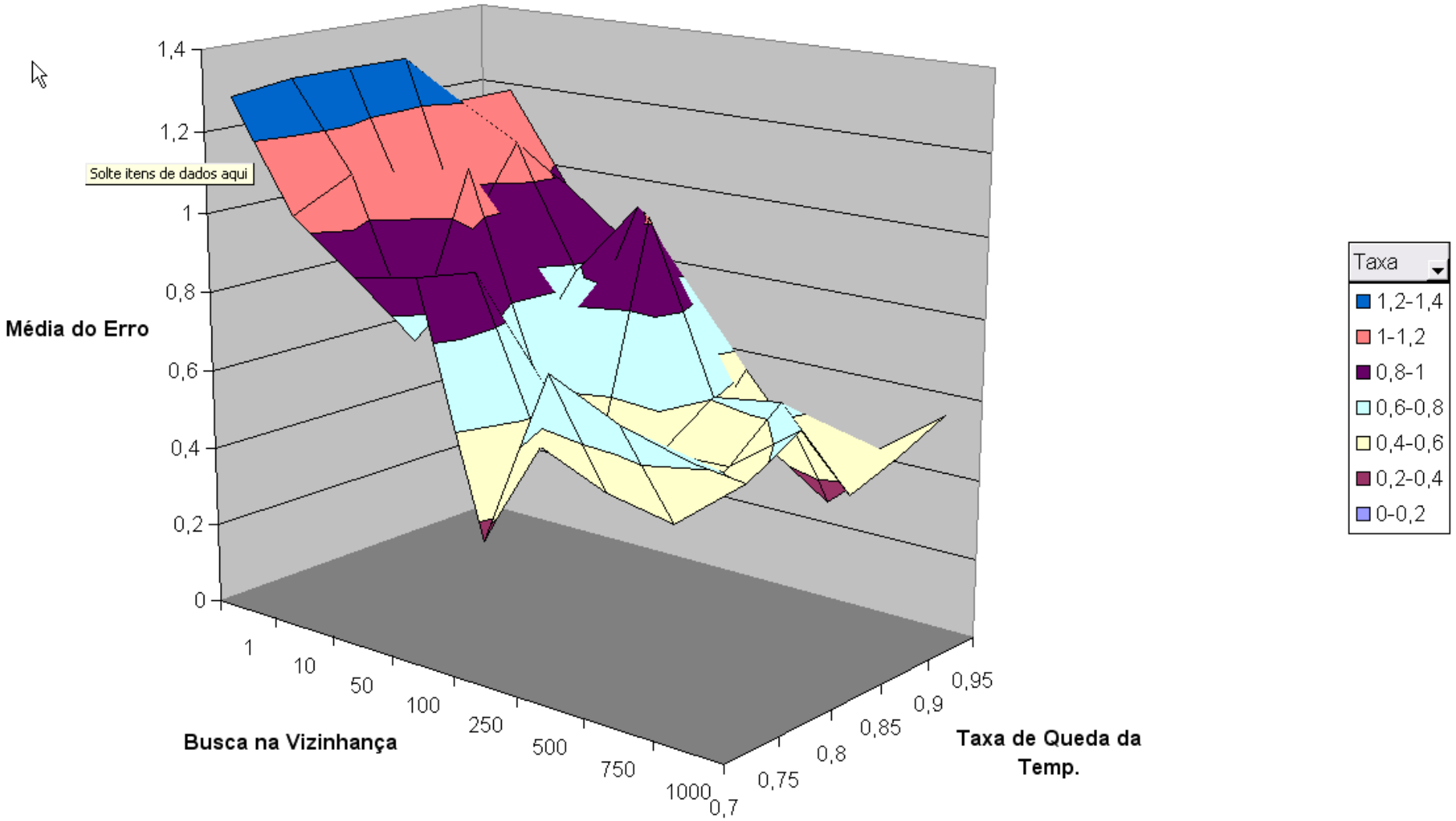
# Parâmetros x Erro (9)

## F-n45-k4

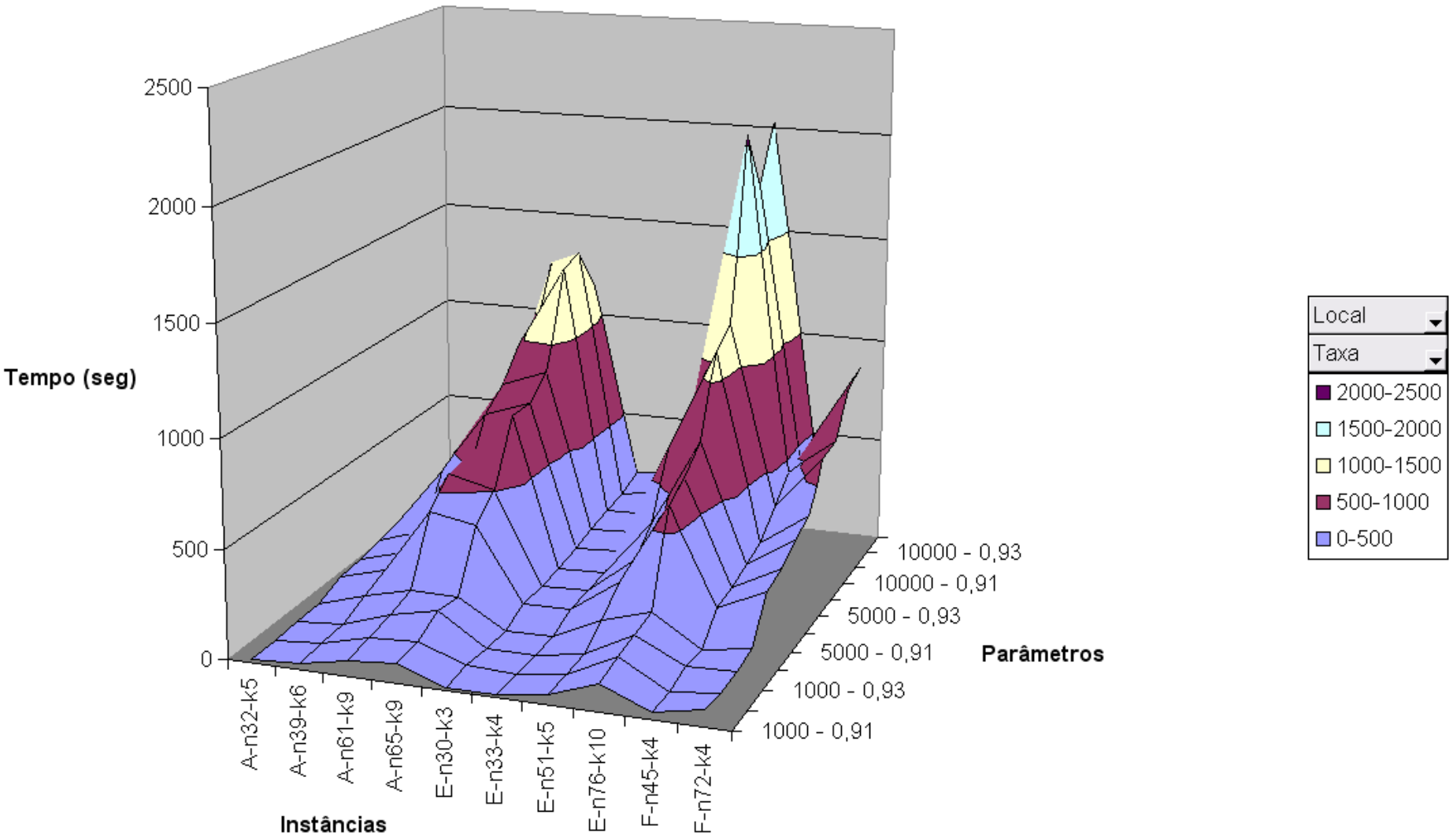


# Parâmetros x Erro (10)

## F-n72-k4



# Parâmetros x Tempo



# Resultados

- Fornecem informações de como os parâmetros influenciam na qualidade da solução;
- Indicam como o tempo de processamento requerido cresce rapidamente com o aumento dos vértices;
- Apesar de existir uma garantia de que com tempo infinito, o Simulated Annealing encontra a melhor solução, para as maiores buscas na vizinhança não se obteve sempre

# Conclusões

- O estudo dos parâmetros é tão importante quanto a heurística a ser empregada;
- Problemas de otimização combinatórios demandam muito tempo de CPU, tornando custoso o estudo dos parâmetros para vários casos de teste;
- Foram gastas mais de 15 horas rodando os casos de teste para diferentes parâmetros.

# Referência

- Material com referência e instâncias do problema
  http://neo.lcc.uma.es/radi-aeb/WebVRP/
- Wikipedia
  http://en.wikipedia.org/wiki/Simulated_annealing
- CMU Artificial Intelligence Repository
  http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/anneal/as
  html