

基于并行人工免疫算法的大规模 TSP 问题求解

戚玉涛, 焦李成, 刘 芳

(西安电子科技大学智能信息处理研究所, 陕西西安 710071)

摘 要: 为求解大规模 TSP 问题, 提出了并行人工免疫系统的塔式主从模型(Towerlike Master2Slave Model, TMSM), 和基于 TMSM 的并行免疫记忆克隆选择算法(Parallel Immune Memory Clonal Selection Algorithm, PIMCSA). TMSM 是粗粒度的两层并行人工免疫模型, 其设计体现了分布式的免疫响应和免疫记忆机制. PIMCSA 用疫苗的迁移代替了抗体的迁移, 兼顾了种群多样性的保持和算法的收敛速度. 与其他算法相比, PIMCSA 在求解精度和运行时间上都更具优势, 而且问题规模越大优势越明显. TMSM 很好地体现了免疫系统的特性, PIMCSA 是适合求解大规模复杂优化问题的并行人工免疫算法, 具有良好的可扩展性.

关键词: TSP; 并行人工免疫系统; 克隆选择; 免疫记忆

中图分类号: TP183 **文献标识码:** A **文章编号:** 03722112 (2008) 08155207

Parallel Artificial Immune Algorithm for Large2Scale TSP

QI Yu2tao, JIAO Li2cheng, LIU Fang

(Institute of Intelligent Information Processing, Xidian University, Xi an, Shaanxi 710071, China)

Abstract: This paper presents a parallel model termed as towerlike master2 slave model (TMSM) for artificial immune systems. Based on TMSM, the parallel immune memory clonal selection algorithm (PIMCSA) is also designed for dealing with large2 scale TSP problems. TMSM is a two level coarse2grained parallel artificial immune model with distributed immune response and distributed immune memory. In PIMCSA, vaccines are extracted and migrated between populations rather than antibodies as has been done in parallel genetic algorithms, it is a good balance between the diversity maintenance of populations and the convergent speed of the algorithm. PIMCSA shows superiority over other compared approaches both in solution quality and computation time, and the larger the problem size the more outstanding the predominance will be. TMSM is a good simulation of biological immune system, and PIMCSA is a parallel artificial immune algorithm with good extensibility, which is capable of solving large scale and complex optimization problems.

Key words: TSP; parallel artificial immune system; clonal selection; immune memory

1 引言

旅行商问题(TSP)是经典的组合优化问题, 具有很强的工程背景和广泛的应用价值. TSP 问题可以形式化描述为: 已知 N 个城市 $C = \{C_1, C_2, \dots, C_N\}$, 以及任意两城市之间的距离 $d(C_i, C_j)$, 求一条经过 C 中所有城市一次且仅一次的闭合路径 $C_P = \{C_{P(1)}, C_{P(2)}, \dots, C_{P(N)}\}$ 使得总行程 $\sum_{i=1}^{N-1} d(C_{P(i)}, C_{P(i+1)}) + d(C_{P(N)}, C_{P(1)})$ 最小^[1].

对于大规模 TSP 问题, 人们倾向于用有限的时间找到可接受的近似解. 求解 TSP 问题的近似算法分为环路构造算法和环路改进算法两类. 环路构造算法从某个非

法解出发, 逐步改变路径, 直到得到一个合法路径为止. 这类算法包括: 最近邻算法, 贪心算法, Clark2Wright 算法, Christofides 算法等^[1]. 环路改进算法则在给定初始合法解之后, 使用某种策略寻找质量更好的解. 这类算法包括: 局部搜索策略(2Opt, LK, IKH, 循环 LK^[2]等), 禁忌搜索^[1], 模拟退火^[1], 遗传算法^[3], 蚁群算法^[4], 粒子群算法^[5], 多级算法^[6,7], 免疫算法^[8]等.

TSP 问题的解空间随着问题规模的增大而迅速膨胀, 面对大规模 TSP 问题庞大的搜索空间, 单个计算机的计算能力已经远不能满足搜索算法对时间的要求. 并行算法求解大规模 TSP 问题越来越受到研究者的关注, 出现了并行蚁群算法^[9,10]研究的一些成果, 目前尚处于起步阶段. 本文工作尝试设计并行的免疫算法来解决这

一复杂问题。

人工免疫系统(Artificial Immune System, AIS)是模仿生物免疫系统的免疫应答、免疫记忆、免疫调节等机理,构造出的一类高性能、自组织、鲁棒性好的人工智能系统。随着人工免疫系统研究的深入,产生了不少的模型和算法,但是模拟生物免疫系统并行性的工作还屈指可数。在网络入侵检测^[11]和分布式机器人控制^[12]的应用中,研究者初步模拟了生物免疫系统并行性,但是这种并行性的尝试是问题性质决定的,目的并非是要提高算法的求解效率。直到对并行克隆选择算法^[13]研究的出现,人们才利用免疫系统潜在的并行性并行化已有的算法,以提高效率。随后出现的 AIRS 并行化算法^[14]进一步说明了并行技术能够有效地应用于人工免疫系统算法,大幅度提高算法的求解能力。生物免疫系统所蕴涵的天然并行性以及高性能并行计算平台的发展,为并行人工系统的研究铺平了道路,也为求解大规模 TSP 问题提供了可行的解决方案。目前,并行人工免疫系统的研究成果多是对已有的人工免疫算法的并行实现,关于并行人工免疫系统模型的研究尚不多见。然而,并行算法并非仅仅是将已有的串行算法用多个处理器并行实现这样简单。

本文借鉴并行遗传算法研究的成功经验,针对人工免疫系统自身的特点设计了求解大规模 TSP 问题的并行人工免疫系统模型,塔式主从模型(Towerlike Master2Slave Model, TMSM),以及基于 TMSM 的并行免疫记忆克隆选择算法(Parallel Immune Memory Clonal Selection Algorithm, PIMCSA)。仿真结果表明,PIMCSA 与当前最有效的局部搜索算法之一循环 LK 算法^[17]和公认性能最好的纯进化类随机搜索算法郭涛算法^[3]相比,求解性能更好和运行速度更快,且具有很好的可扩展性。

2 塔式主从模型

并行遗传算法的粗粒度模型是被广泛关注的一种并行模型。该模型不仅能够加快算法求解大规模复杂问题的速度,而且多种群的搜索能够使得算法更加稳定,不易陷入局部最优^[15]。设计并行人工免疫系统时,不仅要考虑子抗体种群的划分、拓扑结构和交互方式,还要考虑如何模拟生物免疫系统特有的免疫记忆、免疫响应,免疫疫苗等机制。鉴于此,我们设计了粗粒度的两层并行模型,塔式主从模型(Towerlike Master2Slave Model, TMSM)。

定义 1 TMSM 由一个记忆种群 M 和若干个抗体种群 P 按照图 1 中的方式组织而成。

TMSM 的上层是记忆种群 $M = (\hat{A}_1, \hat{A}_2, \dots, \hat{A}_m)$, 由 m 个记忆抗体组成。 M 中的每个记忆抗体 $\hat{A}_i (i = 1, 2, \dots, m)$ 都和一个抗体子种群 $P_i = (A_{i1}, A_{i2}, \dots, A_{in})$ 相对应。因此, TMSM 的下层是 m 个规模为 n 的抗体子种群。

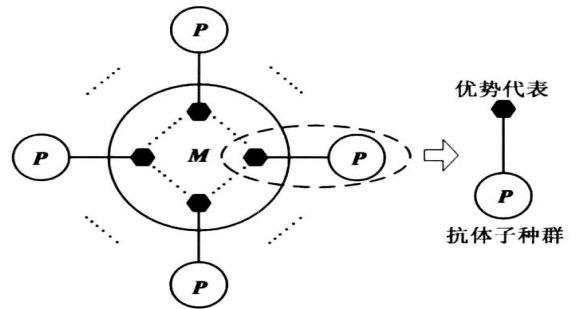


图 1 塔式主从模型

定义 2 若记忆抗体 $\hat{A}_i (i = 1, 2, \dots, m)$ 对应的抗体子种群为 P_i , 称 \hat{A}_i 为 P_i 的优势代表。

TMSM 继承了粗粒度并行模型多种群进化的优点,并引入了记忆种群,借助驱动算法模拟了免疫记忆机制。同时,免疫响应机制以及疫苗的自适应提取和接种机制也能够通过该模型展开。由于记忆种群进程担任了发起和终止计算,以及调度子抗体种群间信息交互的任务,才有了主和从的概念。记忆种群是/主0,抗体子种群是/从0。这与并行遗传算法的主从式并行模型^[15]有着本质的区别。

3 并行免疫记忆克隆选择算法

并行免疫记忆克隆选择算法(Parallel Immune Memory Clonal Selection Algorithm, PIMCSA)包括记忆种群免疫算法和抗体子种群免疫算法两个部分。

算法采用了路径表示(path representation)^[16]的编码方式,抗体 A 的亲合度定义为:

$$\text{Affinity}(A) = (\text{Length}(A) - \text{HKB}) / \text{HKB} \quad (1)$$

$\text{Length}(A)$ 表示抗体 A 解码后的路径长度。HKB 表示 TSP 问题的 HeldKarp Bound^[1], 是对 TSP 最优路径长度的估计。

3.1 记忆种群免疫算法

记忆种群免疫算法的伪码描述如下:

记忆种群免疫算法

```

设定算法终止条件, 令进化代数  $r = 0$ ;
随机初始化记忆抗体种群  $M(r)$  并计算亲合度, 置疫苗集合  $V(r) = S$ ;
While(算法终止条件不满足)
{
    试接收每个  $P_i$  发给  $M(r)$  的抗体  $A_i$ , 若收到且  $A_i$  亲合度大于  $\hat{A}_i$ , 则用  $A_i$  代替  $\hat{A}_i$ ;
    执行记忆成熟操作:  $M(r+1) = \text{Maturation}(M(r))$ ;
    执行动态疫苗提取操作:  $V(r+1) = \text{Extraction}(M(r+1))$ ;
    执行疫苗分配操作:  $\text{Dispatch}(P_1, P_2, \dots, P_m)$ ;
     $r = r + 1$ ;
}
统计并输出结果, 发送算法终止信号到每个子抗体种群。
    
```

记忆成熟操作 记忆成熟操作采用循环 LK 算法^[7]的流程. 对于记忆种群 $M(r)$ 中的每个记忆抗体 \hat{A}_i , 首先做 42Opt 局部搜索, 得到 \hat{A}_c ; 然后用 LK 算法优化 \hat{A}_c , 得到局部最优解 \hat{A}_d . 如果 \hat{A}_d 的亲合度大于 \hat{A}_i , 则用 \hat{A}_d 替代 \hat{A}_i , 否则仍然保留 \hat{A}_i . 记忆成熟操作后的种群为 $M(r+1)$.

动态疫苗提取操作 首先, 将当前记忆抗体种群 $M(r+1)$ 分成两个抗体集合: 疫苗抽取集合 $M_1(r+1)$ 和疫苗接种集合 $M_2(r+1)$. 令 k 为小于等于 $m/2$ 的最大正整数, 从 $M(r+1)$ 中随机选择 k 个记忆抗体构成 $M_1(r+1)$, 剩下的抗体构成 $M_2(r+1)$. 然后, 对集合 $M_1(r+1)$ 中所有的记忆抗体进行求交集运算, 得到 k 个路径的公共边集合 $E(r+1)$. 将 $E(r+1)$ 中有公共城市的边合并成公共子路径, 将得到的公共子路径和剩下的公共边分别作为多基因疫苗组和单基因疫苗存储到疫苗集合 $V(r+1)$ 中. 在 $V(r+1)$ 中, 单基因疫苗代表了路径中的一条边, 存储形式是边两端的城市序列, 疫苗长度为 1. 疫苗组代表了一段子路径, 即若干条首尾相接的边的序列, 存储形式是若干城市的有序排列, 疫苗长度为子路径中包含的边的数量.

疫苗分配操作 在疫苗集合 $V(r+1)$ 中任选一个疫苗 v_i , 可能是单基因疫苗或者多基因疫苗组. 然后在疫苗接种集合 $M_2(r+1)$ 中任选一个记忆抗体 \hat{A}_j . 将疫苗 v_j 发送给 \hat{A}_j 所对应的抗体子种群 P_j .

3.1.2 抗体子种群免疫算法

运行在抗体子种群 $P_i (i=1, 2, \dots, m)$ 上的免疫算法的伪码描述如下:

抗体子种群免疫算法

令进化代数 $t=0$, 置终止信号量 $Halt=False$;

随机初始化记忆抗体种群 $P_i(t) = (A_i^1(t), A_i^2(t), \dots, A_i^n(t))$ 并计算亲合度;

记 $P_i(t)$ 中的最优抗体为 $A_i(t)$, $CurrentBest = Affinity(A_i(t))$, 发送 $A_i(t)$ 到 M ;

置当前疫苗集合 $V(t) = 5$;

While ($Halt$ 不为 True)

{
 试接收记忆种群 M 发来的算法终止信号, 若收到则置 $Halt=True$, 跳出循环;

 试接收记忆种群 M 发来的疫苗, 若收到则用新的疫苗代替 v ;

 执行克隆操作: $P_c(t) = CL(P_i(t))$;

 执行免疫基因操作: $P_d(t) = IG(P_c(t))$;

 执行克隆选择操作: $P_i(t+1) = SL(P_d(t))$;

 找到 $P(t+1)$ 中的最优抗体 $A_i(t+1)$, 若 $Affinity(A_i(t+1)) > CurrentBest$, 则置 $CurrentBest = Affinity(A_i(t+1))$ 并发送 $A_i(t+1)$ 到 M , $t = t + 1$;

}

$$CL(P_i(t)) = (CL(A_i^1(t)), CL(A_i^2(t)), \dots, CL(A_i^n(t))) \quad (2)$$

其中, $CL(A_j^i(t)) = I_j @ A_j^i(t) (j=1, 2, \dots, n, I_j$ 为 q_j 维单位行向量) 称抗体 $A_j^i(t)$ 的 q_j 克隆. 克隆规模 q_j 与抗体 $A_j^i(t)$ 的亲合度有关, 亲合度越大 q_j 越大.

令 $Y_c(t) = CL(A_j^i(t)) = \{y_{c1}(t), y_{c2}(t), \dots, y_{cq}(t)\}$ 则 CL 操作后的抗体子种群可记为:

$$P_c(t) = \{Y_c(t), Y_c(t), \dots, Y_c(t)\} \quad (3)$$

免疫基因操作 免疫基因操作定义如下,

$$IG(P_c(t)) = (IG(Y_c(t)), IG(Y_g(t)), \dots, IG(Y_c(t))) \quad (4)$$

记 $Y_c(t) = IG(Y_c(t)) = \{y_{d1}(t), y_{d2}(t), y_{dq}(t)\}$, 则 IG 操作后的抗体子种群可记为:

$$P_d(t) = \{Y_d(t), Y_d(t), \dots, Y_d(t)\} \quad (5)$$

免疫基因操作 IG 以等概率在疫苗接种算子 $Vacc2$ 和 $Inver2over$ 算子^[3] 中任选其一作用在抗体之上. 对抗体 A 接种疫苗的操作流程如下, 接种后的抗体为 Ac :

疫苗接种算子

Vaccination(A)

{

 If (抗体子种群的当前疫苗 v 不为空)

 {

$Ac = A$, c 为 v 中的第一个城市, α 为 v 中 c 的下一个城市;

 While(c 不是 v 中的最后一个城市)

 {

 翻转 Ac 中 c 的下一个城市和 α 之间的城市;

$c = \alpha$, $\alpha = v$ 中 α 的下一个城市;

 }

 }

}

$Inver2over$ 算子是在抗体周围的编码空间进行局部搜索, 利用了抗体子种群内部的启发信息. 疫苗接种算子将记忆种群学习到的知识引入抗体之中, 利用了来自其他抗体子种群的启发信息.

克隆选择操作 克隆选择操作定义如下,

$$SL(P_d(t)) = (SL(Y_d(t)), SL(Y_d(t)), \dots, SL(Y_d(t))) \quad (6)$$

若 $A_j^i(t+1) = SL(Y_d(t)) (j=1, 2, \dots, n)$, 则 SL 操作后的种群为:

$$P_i(t+1) = (A_i^1(t+1), A_i^2(t+1), \dots, A_i^n(t+1)) \quad (7)$$

作用在 $Y_c(t)$ 上的 SL 操作 $SL(Y_d(t))$ 流程如下: 在 $Y_d(t)$ 中选择抗体亲合度最大的抗体, 记为:

$$y_d(t) = \{y_{dk}(t) | \max \text{affinity}(y_{dk}(t)), k=1, \dots, q\} \quad (8)$$

若 $Affinity(y_d(t)) > Affinity(A_j^i(t))$, 则令 $SL(Y_d(t)) = y_d(t)$, 否则 $SL(Y_d(t)) = A_j^i(t)$.

克隆操作 克隆操作 CL 定义如下,

4 算法性能分析

PIMCSA 不是对已有免疫算法的并行实现, 而是建立在并行人工免疫系统模型 TMSM 的基础之上. PIMCSA 继承了并行遗传算法粗粒度模型的优点, 同时引入了免疫记忆, 免疫疫苗等免疫系统机制. PIMCSA 将抗体种群划分为抗体子种群和记忆种群两类, 对前者的进化采用了计算代价较小的免疫基因操作对解空间进行粗放式搜索; 对后者的进化采用了启发性强的循环 LK 局部搜索对记忆抗体行进行细化搜索. 利用抗体子种群的相对独立性, 记忆种群的多样性能够很好地保持, 而记忆种群提取的疫苗又促进了抗体子种群的进化.

PIMCSA 中各个抗体子种群之间通过疫苗的迁移来交换信息, 而非像粗粒度并行遗传算法那样依靠个体的迁移. 这是本文工作的重要创新之处. 这样做有三方面的优势: 其一, 疫苗的信息量要远比整个抗体小, 传递疫苗能够节省通信代价. 这种优势问题规模越大就越明显. 其二, 疫苗是由多个抗体子种群所对应的记忆抗体共同产生的(这其中不包括接收疫苗的子种群对应的记忆抗体), 是多个抗体子种群进化经验的总结. 个体迁移策略不区分迁移个体中基因的优劣, 而是通过迭代进化让优秀的基因在种群中自然增殖, 浪费了计算资源. 疫苗迁移策略则直接提取优秀的基因传递给其他子种群, 这种启发信息可用性更强. 文献[7]分析了用 LK 算法得到的局部最优解和全局最优解之间的关系, 发现对局部最优解进行简单的求交集操作能以很高的准确度对全局最优解中出现的边进行预测. 这一研究成果说明 PIMCSA 提取的疫苗是对最优抗体基因的有效估计. 其三, 在个体迁移策略中优秀个体有可能在子种群间多次传递而导致多样性损失, 选择疫苗迁移能够有效解决这一问题.

综上所述, PIMCSA 不仅是可以在多个处理器上的并行免疫算法, 而且在算法的设计中融入了进化经验学习和种群多样性保持机制, 兼顾了算法收敛速度和求解质量的平衡.

5 仿真对比实验与算法有效性分析

本文用 C+ MPI 实现了 PIMCSA. 算法运行在西安电子科技大学智能信息处理研究所和惠普联合实验室的 HPC 集群机上. 集群有 1 个管理节点(AMD Opteron 2. 2GHz 处理器, 4G 内存), 1 个 I/O 节点(2 @AMD Opteron 2. 2GHz 处理器, 4G 内存)和 64 个常规计算节点(2 @AMD Opteron 1. 8GHz 处理器, 4G 内存). 操作系统为均 Red Hat Enterprise Linux AS release 4. 仿真实验使用了

5 个计算节点, 对 TSPLAB^[18] 中对称和非对称 TSP 问题的几个典型实例进行了测试.

我们将 PIMCSA 与循环 LK 算法^[17] (ILK) 和郭涛算法^[3] (GT) 进行了比较. ILK 和 GT 是串行算法, 在 I/O 节点上运行. PIMCSA 的抗体子种群个数 m 置为 8, 抗体子种群规模 n 置为 30. ILK 和 GT 算法的种群规模均置为 30. 三个算法的终止条件设置为: 记忆种群的当前最优路径长度小于等于已知最优路径长度 Opt , 或者当前最优路径在 50 次迭代内保持不变时算法终止. 实验中的数据为算法独立运行 20 次的统计结果.

5.1 对称 TSP 问题

表 1 给出了三种算法对不同规模对称 TSP 问题的仿真实验统计结果. 表中, Percent over Opt 表示算法获得的最优解路径长度和 Opt 的差占 Opt 的百分比, Running Time 的单位为秒.

从表 1 可以看出, ILK 能够获得质量较高的 TSP 解路径, 但是每次迭代的代价过大, 因此运行时间长; GT 每次迭代的时间代价小, 然而容易陷入局部最优, 因而运行时间稍短, 但是得到的 TSP 解路径质量不高. PIMCSA 综合了两者的长处, 在求解质量上有明显改进. 从计算时间上看, 虽然 PIMCSA 使用了 5 个处理器进行计算,

表 1 对称 TSP 问题仿真实验结果比较

Instance	Cities	Opt	Percent over Opt (mean)			Running Time (mean)		
			ILK	GT	PIMCSA	ILK	GT	PIMCSA
ATT532	532	27686	0.086	0.114	0	127.5	83.5	17.4
GR666	666	294358	0.061	0.176	0.003	292.1	102.4	59.6
DSJ1000	1000	18659688	0.133	0.152	0.008	418.6	372.4	52.2
PR2392	2392	378032	0.142	0.357	0.006	102.5	87.3	16.6
RL5915	5915	565530	0.163	0.879	0.047	293.7	226.9	239.1
PLA7397	7397	23260728	0.059	0.356	0.007	8843.2	8221.1	1762.4
RL11849	11849	923288	0.191	0.824	0.105	6311.3	5352.8	2581.3
USA13509	13509	19982859	0.163	1.209	0.067	10352.3	8931.5	3520.7
PLA33810	33810	66050499	0.184	1.813	0.152	76315.8	53356.6	18137.4
PLA85900	85900	142382641	0.246	1.115	0.214	214307.5	113755.9	29883.2
Average			0.1428	0.6995	0.0609	31736.45	19049.04	5626.99

表 2 非对称 TSP 问题仿真实验结果比较

Instance	Cities	Opt	Percent over Opt (mean)			Running Time (mean)		
			ILK	GT	PIMCSA	ILK	GT	PIMCSA
BR17	17	39	0	0	0	2.14	3.57	0.12
P43	43	5620	0.082	0.372	0.002	33.2	21.7	25.3
RY48P	48	14422	0.037	0.506	0.015	23.6	24.2	1.34
FTV70	71	1950	0.132	0.358	0.039	42.9	38.1	2.33
KROI24P	100	36230	0.031	0.114	0.007	72.3	55.6	1.15
FT53	106	6905	0	0.037	0	31.2	63.4	1.85
FTV170	171	2755	0.018	0.326	0.044	69.3	42.7	1.03
RBG358	358	1163	0.139	1.977	0.006	137.5	114.6	36.6
RBG403	403	2465	0.082	0.973	0	291.7	174.3	59.4
RBG443	443	2720	0.074	0.661	0	372.5	351.1	84.9
Average			0.0595	0.5324	0.0113	107.634	88.927	21.402

但是 ILK 的平均计算时间为 PIMCSA 的 5.6 倍. 虽然 GI 的平均计算时间仅为 PIMCSA 的 3.4 倍, 然而求解质量不高. 另外 I/O 节点的计算能力要优于普通计算节点. 因此 PIMCSA 在求解路径质量和计算时间上都取得了很好的效果.

512 非对称 TSP 问题

对于城市数为 N 的非对称 TSP 问题, 我们利用 Jonker 和 Volgenant 设计的方法转换成城市数为 $2N$ 的对称 TSP 问题求解^[19].

若 $C = [c_{ij}]_{N \times N}$ 为非对称 TSP 问题的代价矩阵, 那么转换成对称 TSP 问题的代价矩阵 $Cc = [c_{ij}']_{2N \times 2N}$ 的计算方法如式(9)所示, 其中 L 是一个足够大的实数, 本文取 $L = \max(c_{ij})$.

$$\begin{aligned}
 c_{i,j} &= c_{j,N+i} = c_{ij}, & i, j = 1, 2, \dots, N \text{ and } i \neq j \\
 c_{i,N+i} &= c_{j,N+i} = -L, & i = 1, 2, \dots, N \\
 c_{ij} &= L, & \text{otherwise}
 \end{aligned} \tag{9}$$

表 2 给出了三种算法对非对称 TSP 问题的仿真实验统计结果. 表中各项的意义与表 1 相同. 从表中的数据我们可以得到和对称 TSP 问题的实验相同的结论.

513 疫苗提取策略的有效性分析

该实验验证 PIMCSA 疫苗提取策略的有效性. 图 2 和图 3 用对称 TSP 和非对称 TSP 实例考察了 PIMCSA 疫苗提取策略所提取的边在最优路径中出现的比例. 比例越高说明疫苗提取策略对最优解的估计越准确, 对应的疫苗提取策略效果越好.

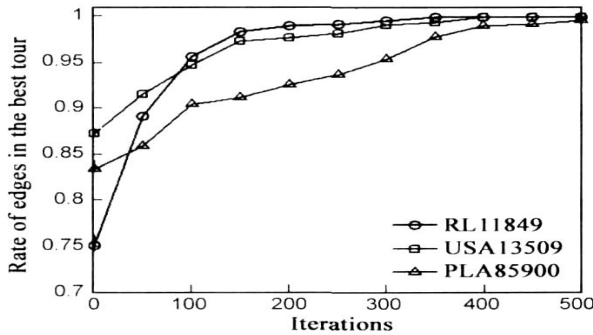


图 2 对称 TSP 实例疫苗预测精度

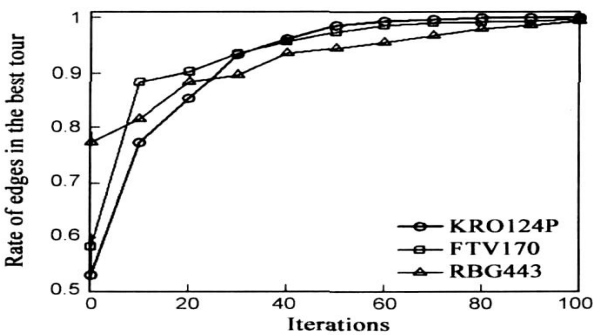


图 3 非对称 TSP 实例疫苗预测精度

观察图中的曲线可以看出, PIMCSA 提取的疫苗对最优解中路径预测的准确率随着迭代的进行迅速提高, 并逐渐接近 1. 这说明疫苗的启发信息是有效的, 有助于加速算法的收敛. 进一步观察发现, 问题规模越大, 疫苗的初始预测精度越高, 随着迭代的进行精度提高越缓慢; 问题规模较小时, 疫苗的初始预测精度越低, 但是随着迭代的进行精度迅速提高.

514 并行算法可扩展性分析

加速比就是一个衡量并行解题过程中的相对收益的指标. 并行系统的加速比是指对于一个给定的应用, 并行算法的执行速度相对于串行算法的执行速度加快了多少倍. 若 T_s 表示在串行计算机上算法从开始到结束所需要的时间, T_p 表示并行计算机上算法从开始到最后一个处理器完成任务所需要的时间, 那么加速比 S 定义为:

$$S = \frac{T_s}{T_p} \tag{10}$$

效率被用来衡量一个处理器的计算能力被有效利用的比率. 若处理器个数为 p , 则效率 E 定义为:

$$E = \frac{S}{p} \tag{11}$$

若 W 为问题总的计算量, $T_0(W, p)$ 为额外开销 (是 W 和 p 的函数), T_p 可以表示为:

$$T_p = \frac{W + T_0(W, p)}{p} \tag{12}$$

因而, 加速比 S 和效率 E 可以表示为:

$$S = \frac{W}{T_p} = \frac{pW}{W + T_0(W, p)} = \frac{p}{1 + T_0(W, p)/W} = \frac{p}{1 + \delta} \tag{13}$$

$$E = \frac{S}{p} = \frac{1}{1 + T_0(W, p)/W} = \frac{1}{1 + \delta} \tag{14}$$

可见, δ 是影响算法加速比和处理器效率的关键. 在问题规模一定 (W 不变) 的情况下, 额外开销代价越小 δ 就越小, 算法的加速比和处理器效率就越高. 当 W 一定, $T_0(W, p)$ 就只与 p 相关, 可以记为 $T_0(p)$, 这个函数是由并行算法决定的. 因此研究不同问题规模下 δ 的变化情况就是探讨并行算法的通信代价随着处理器个数增加而增加的情况, 即 $T_0(p)$ 随 p 增加而变化的情况. $T_0(p)$ 上升得越慢, 算法可扩展性越好, 反之则越差. 由式(13)可以得到:

$$\delta = \frac{T_0(W, p)}{W} = \frac{p}{S} - 1 = \frac{1}{E} - 1 \tag{15}$$

图 4 和图 5 用对称 TSP 和非对称 TSP 实例考察了 PIMCSA 的 δ 随着处理器个数 p 的增加而变化的情况. 实验中, 抗体子种群规模为 30, 记忆种群规模为子种群个数, 即 $p-1$ 个. 图中数据为算法独立运行 20 次的统计结果.

从图中可以看出, 当问题规模一定时 δ 随着处理

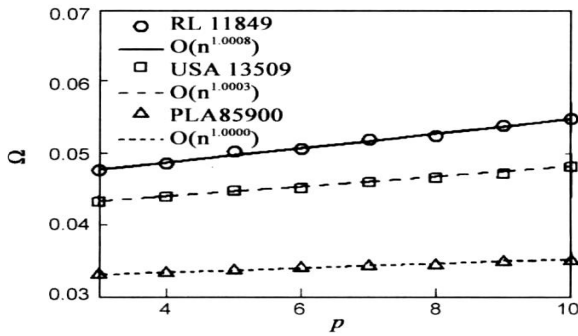


图 4 对称 TSP 实例的可扩展性

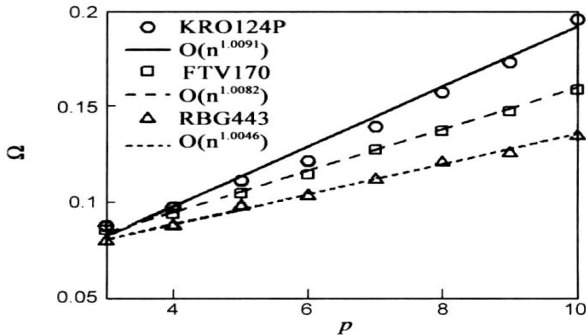


图 5 非对称 TSP 实例的可扩展性

器个数 p 的增加大致呈线性递增的趋势, 这说明 PIMCSA 具备很好的可扩展性. 问题规模越大, θ 的值越小, 增长越缓慢, 这说明对于大规模的 TSP 问题 PIMCSA 的可扩展性更好.

这样的实验结果是合理的. PIMCSA 中额外开销包括三部分: 抗体子种群将最优抗体发送给记忆种群以及记忆种群接收抗体的通信开销 (记为 C_1), 记忆种群产生疫苗的开销 (记为 C_2) 和疫苗传递的开销 (记为 C_3). C_1 在问题规模一定的条件下随着 p 的增加线性递增, C_2 是线性复杂度的求交集操作, 同样随着 p 的增加线性递增, C_3 与 p 的值无关. C_1 和 C_2 是主要的额外开销, 对于大规模的 TSP 问题 C_3 相对于 C_1 和 C_2 可以忽略不计. 因此, 额外开销在问题规模一定的条件下随着 p 的增长大致呈线性递增关系, 而且问题规模越大越接近线性. TSP 问题的规模越大, 问题总的计算量 W 越大, 因而 θ 的增长速度越缓慢.

6 总结与展望

为求解大规模 TSP 问题, 本文提出了一种并行的人工免疫系统模型 TMSM, 以及基于 TMSM 的驱动算法 PIMCSA. 与其他并行免疫算法的研究成果不同, PIMCSA 不是对已有免疫算法的并行化实现, 而是具备了抽象的并行化模型的人工免疫系统.

TMSM 的设计体现了分布式的免疫识别和免疫响应机制, 抗体子种群和记忆抗体之间的对应关系又带来了分布式的免疫记忆. 基于 TMSM 的 PIMCSA 兼顾了抗

体子种群的粗放式搜索和记忆种群的细化搜索. 抗体子种群相对独立的进化保持了很好的种群多样性, PIMCSA 又将抗体子种群的多样性引入记忆种群之中, 使得算法不易陷入早熟收敛. PIMCSA 针对 TSP 问题设计了疫苗提取和接种策略, 更具特色的是 PIMCSA 用疫苗的迁移代替了抗体的迁移, 这样做不仅降低了通信代价而且在疫苗的提取过程中融入了进化经验的学习机制, 使得传递给抗体子种群的信息可用性更强, 提高了通信的效率. 总之, TMSM 是体现了免疫系统特性的并行模型, PIMCSA 是适合运行在并行系统之上的人工免疫算法.

对 TSPLIB 中对称和非对称的 TSP 问题的仿真实验结果表明, PIMCSA 无论在求解精度还是在运行时间上都有很好的表现. 我们还验证了 PIMCSA 疫苗提取策略的有效性, 实验结果表明, 疫苗对最优解中路径预测的准确率随着迭代的进行迅速提高, 并逐渐接近 1. 另外, 作者还从理论上分析了并行算法的加速比和处理器的效率与变量 θ (算法的额外开销和问题总的计算量的比值) 相关. 实验结果显示, PIMCSA 的随着处理器个数 p 的增加大致呈线性递增的趋势, 说明 PIMCSA 具备很好的可扩展性.

PIMCSA 是适合求解大规模复杂优化问题的并行人工免疫算法. 如何设计基于 TMSM 的算法解决其他问题, 以及研究抗体子种群规模、数量和处理器计算能力之间的关系是进一步研究的重要方向.

参考文献:

- [1] D S Johnson, L A McGeoch. The traveling salesman: a case study in local optimization[A]. Local Search in Combinatorial Optimization[C]. New York: Wiley and Sons, 1997. 215 - 310.
- [2] D S Johnson, L A McGeoch. Experimental analysis of heuristics for the STSP[A]. The Traveling Salesman Problem and its Variations[C]. Boston: Kluwer Academic Publishers, 2002. 369 - 443.
- [3] T Guo, Z Michalewicz. Inve2over operator for the TSP[A]. Proceeding of the 5th Parallel Problem Solving form Nature [C]. Berlin: springer, 1998. 803- 812.
- [4] X M Song, B Li, H M Yang. Improved ant colony algorithm and its applications in TSP[A]. Proceedings of the 6th International Conference on Intelligent Systems Design and Applications[C]. Washington DC: IEEE Computer Society, 2006. 1145 - 1148.
- [5] X X He, et al. A new algorithm for TSP based on swarm intelligence[A]. Proceedings of the 6th World Congress on Intelligent Control and Automation [C]. Dalian, China: IEEE, 2006. 3241 - 3244.
- [6] C Walshaw. A Multilevel Lin2Kernighan2Helsgaun Algorithm

- for the Travelling Salesman Problem[R]. London: University of Greenwich, 2001.
- [7] 邹鹏, 周智, 陈国良, 顾均. 求解 TSP 问题的多级归约算法[J]. 软件学报, 2003, 14(1): 35- 42.
Zou Peng, Zhou zhi, Chen Guoliang, Gu Jun. A multilevel reduction algorithm to TSP[J]. Journal of Software, 2003, 14(1): 35- 42. (in Chinese)
- [8] 王磊, 潘进, 焦李成. 免疫规划[J]. 计算机学报, 2000, 23(8): 806- 812.
Wang Lei, Pan Jin, Jiao Licheng. The immune programming[J]. Chinese Journal of Computers, 2000, 23(8): 806- 812. (in Chinese)
- [9] 吕强, 高彦明, 钱培德. 共享信息素矩阵: 一种新的并行 ACO 方法[J]. 自动化学报, 2007, 33(4): 418- 421.
Lv Qiang, Gao Yanming, Qian Peide. Sharing one pheromone matrix: a new approach to parallel ACO[J]. Acta Automatica Sinica, 2007, 33(4): 418- 421. (in Chinese)
- [10] Chen Ling, Zhang Chunfang. Adaptive exchanging strategies in parallel ant colony algorithm[J]. Journal of Software, 2007, 18(3): 617- 624.
- [11] J W Kim. Integrating Artificial Immune Algorithms for Intrusion Detection[D]. London: Department of Computer Science, University College London, 2002.
- [12] H Y Lau, V W Wong. Immunologic control framework for automated material handling[A]. Proceeding of the 2nd International Conference on Artificial Immune Systems[C]. England: Springer, 2003. 57- 68.
- [13] A Watkins, X Bi, A Phadke. Parallelizing an immune inspired algorithm for efficient pattern recognition[A]. Intelligent Engineering Systems through Artificial Neural Networks, Vol. 14, Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Complex Systems and Artificial Life[C]. New York: ASME Press, 2003. 225- 230.
- [14] A Watkins, J Timmis. Exploiting parallelism inherent in AIRS, an artificial immune classifier[A]. Proceeding of the 3rd International Conference on Artificial Immune Systems[C]. Italy: Springer, 2004. 427- 438.
- [15] Erick Cant Paz. Efficient and Accurate Parallel Genetic Algorithms[M]. Boston: Kluwer Academic Publishers, 2000.
- [16] Z Michalewicz, 著, 曹宏庆, 等, 译. 如何求解问题: 现代启发式方法[M]. 北京: 中国水利出版社, 2003: 149- 154.
- [17] D S Johnson. Local optimization and the traveling salesman problem[A]. Proceedings of the 17th Colloquium on Automata, Language, and Programming[C]. Berlin: Springer, 1990. 446- 461.
- [18] G Reinelt. TSPLIB: A traveling salesman problem library[J]. ORSA Journal on Computing, 1991, 3(4): 376- 384.
- [19] R Jonker, T Volgenant. Transforming asymmetric into symmetric traveling salesman problems[J]. Operations Research Letters, 1983, 2: 161- 163.

作者简介:



戚玉涛 男, 1981年出生于河南潢川, 西安电子科技大学电子工程学院博士生. 主要研究方向包括: 进化算法、人工免疫系统、分布式计算、机器学习等. E-mail: qi_yutao@163.com



焦李成 男, 1959年10月生于陕西白水, 西安电子科技大学电子工程学院教授, 博士生导师, IEEE高级会员. 主要研究方向包括: 信号和图像处理、非线性电路与系统理论、智能信息处理、数据挖掘和人工免疫系统等.