

基于蚁群优化算法的云数据库动态路径规划

史恒亮^{1,2} 白光一^{1,3} 唐振民¹ 刘传领¹

(南京理工大学计算机学院 南京 210094)¹ (河南科技大学电信学院 洛阳 471003)²

(方舟信息技术有限公司 苏州 215021)³

摘要 云计算是下一代计算网络模型的发展趋势。云中的存储资源如何快速地路由,更是研究的难点。蚁群算法是基于群体的仿生优化算法,具有智能搜索、全局优化、鲁棒性、分布式计算和容易与其他算法相结合等优点。根据这两种事物的特点,提出了合理的结合算法,该算法能够在云中快速、合理地找到所需访问的数据库,减少云数据库路由的动态负荷,从而很大程度上提高云计算的效率。

关键词 蚁群优化算法,云计算,云数据库,动态路径规划

中图法分类号 TP311 文献标识码 A

Cloud Database Dynamic Route Scheduling Based on Ant Colony Optimization Algorithm

SHI Heng-liang^{1,2} BAI Guang-yi^{1,3} TANG Zhen-min¹ LIU Chuan-ling¹

(School of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, China)¹

(School of Electronic Information, Henan University of Science & Technology, Luoyang 471003, China)²

(Noah It Solution Com., LTD, Suzhou 215021, China)³

Abstract Cloud computing is the development trend of next generated computing network model. And how to effectively route storage resource in cloud is development's difficulty in ring of industry. Ant colony algorithm is a bionics optimization algorithm based group ants which has many priorities such as intelligent routing, overall optimization, robustness, distributed computing, ability to mix itself with other algorithm. Based on the above two factors, this paper proposed a reasonable algorithm which can find the requiring database rapidly and effectively, and reduce the dynamic routing burdens of cloud database routing, and enhance the efficiency of cloud computing at a large.

Keywords ACO(ant colony optimization), Cloud computing, Cloud database, Dynamic routing scheduling

云是一种新兴的计算模型,是集分布式操作系统、分布式数据库、网络计算等技术于一体的计算机网络模型,它可以充分地利用云上的硬件资源和软件资源。云数据库作为云中的存储资源,一直是人们研究的难点。云数据库与传统的分布式数据库、静态数据库有着很大的不同,如何在云中快速合理地找到最适合应用的数据库,更是人们研究的难点。

1 云数据库

云数据库系统由若干个站点集合而成。这些站点又称为节点,它们在通讯网络中联接在一起,每个节点都是一个独立的数据库系统,它们都拥有各自的数据库、中央处理机、终端,以及各自的局部数据库管理系统。因此云数据库系统可以看作是一系列集中式数据库系统的联合^[1]。它们在逻辑上属于同一系统,但在物理结构上是分布式的。云数据库系统已经成为信息处理学科的重要领域,正在迅速发展之中,原因基于以下几点:

(1) 它可以解决组织机构分散而数据需要相互联系的问题。比如银行系统,总行与各分行处于不同的城市或城市中

的各个地区,这就需要分布式的系统。

(2) 如果一个组织机构需要增加新的相对自主的组织单位来扩充机构,则云数据库系统可以在对当前机构影响最小的情况下进行扩充。

(3) 均衡负载的需要。数据的分解采用使局部应用达到最大,这使得各处理机之间的相互干扰降到最低。负载在各处理机之间分担,可以避免临界瓶颈。(4) 当现有机构中已存在几个数据库系统,而且实现全局应用的必要性增加时,就可以由这些数据库自下而上构成云数据库系统。

(5) 相等规模的云数据库系统在出现故障的几率上不会比集中式数据库系统低,但由于其故障的影响仅限于局部数据应用,因此就整个系统来讲它的可靠性是比较高的。

以上几点说明云数据库是分布的、动态的、全局均衡的、可伸缩性的数据库。

2 蚁群算法

2.1 蚁群算法

蚁群算法是新近发展的基于群体智能的仿生优化算法,

到稿日期:2009-06-15 返修日期:2009-09-13 本文受国家自然科学基金项目(90820306)资助。

史恒亮(1976—),男,博士生,讲师,CCF会员,主要研究领域为嵌入式数据库、嵌入式系统仿真,E-mail:shi_hl@sina.com.cn;白光一(1963—),男,教授,博士生导师,主要研究领域为嵌入式数据库、嵌入式系统仿真;唐振民(1961—),男,教授,博士生导师,主要研究领域为移动机器人、多机器人系统;刘传领(1970—),男,副教授,主要研究领域为算法研究、智能识别。

它模拟蚂蚁的觅食行为来解决复杂的组合优化问题。蚁群算法的优点是智能搜索、全局优化、鲁棒性、分布式计算和容易与其他算法相结合等^[2]。

蚁群算法是受到对真实蚁群行为研究的启发而提出的。蚂蚁在外出觅食时,个体之间通过一种称为外激素(“信息素”)的物质进行信息传递,蚂蚁在运动过程中,能够在经过的路径上留下这种物质,并能感知这种物质,并依次指导自己的运动方向。大量蚂蚁组成的蚁群的集体行为便体现为一种信息的正反馈现象,即某一路径上走过的蚂蚁越多,留下的信息就越多,后者选择该路径的可能性就越大。

2.2 算法特征

A)每隔一定的时间间隔,与数据流的传输并发,在每个网络结点中,人工蚂蚁根据流量分布选择目的结点并异步地向这些结点移动。

B)人工蚂蚁之间既协同工作又相互独立。它们通过在本结点中读写信息素,以间接的途径相互交流。

C)每个人工蚂蚁都搜索一条连接它的源结点和目的结点并且代价最小的路径。

D)所有的人工蚂蚁都是一步接一步地向目的结点前进的。在每个中间结点上,蚂蚁都使用一个贪婪随机策略来选择下一个要到达的结点。

E)在移动过程中,人工蚂蚁收集以下信息,包括时间长度、拥塞状态以及所经过的路径结点标识符。

F)蚂蚁一旦到达目的结点就沿着同样的路径朝相反的方向返回源结点。

G)在返回源结点的过程中,人工蚂蚁使用途径结点的路径质量为变量构造函数,调整网络状态的本地结点模型以及所访问过的结点中的信息素。

H)人工蚂蚁一旦返回源结点就会被系统删除。

3 问题描述

云是一个规模极其庞大的资源池,池中有极其丰富的计算资源、存储资源。云数据库也是由海量的数据库群所构成,而这些数据库一部分固定存储在云中,还有一部分由中小服务商的服务器构成,而这部分服务器会随着公司的运行情况不断加入到云中,或退出云^[1]。云上的结点与链路有时候会突然失效,而新的结点与链路也可能在任意时刻加入到云中来,因此这就给云中的路由预测和识别带来了很大的难度,这部分就构成了云的动态数据存储资源。

如何按照用户的查询,快速地从云数据库中找到自己所需的资源?规划的目的是使查询由起点 g_{begin} 安全避碰地沿一条最优的路径到达终点 g_{end} 。把这个查找过程形象地看作一个蚂蚁 ANT 来寻找食物的过程,当找到食物后,蚂蚁将从目的结点返回源结点。可以把不同的结点看作是云中的不同的云数据库。

查找的过程是结合了权限管理、流量分发、拥塞控制部件以及路由表的复杂过程。路由表是所有路由算法中的一个关键元素,它记录了算法所要使用的信息,而算法就是根据这些信息做出本地转发决策的。与传统的集中式数据库存储和静态存储数据库不同,云数据库特性决定了它需要一种能够兼顾分布式和动态更新的数据存储方式,而与之相对应的树型查找算法和静态查找算法是不适合云数据库的。确定性算法

可以保证在规模有限的组合优化问题中找到最优解,但当云数据库的规模在无限增大时,找到最优解的可能性就比较小,唯一可行的办法就是降低最优值的精度以换取计算效率的提高,这就需要非确定算法来找到近似解。而云数据库的路径规划问题也正是这类问题中的组合优化问题^[6]。

云数据库是一个规模庞大的网络数据库,设计查找算法时要从全局的观点考虑到网络的流量、拥塞、到达时间、局部最优等问题。传统的最快速梯度下降(rapides gradient-descent)和模拟退火(simulated annealing)算法容易陷入到局部最优的困境中,找不到全局的最优解^[3]。

蚁群算法是一种针对 NP 问题的离散优化问题的元启发式算法,它能够解决静态的组合优化问题,又能够解决动态的组合优化问题,对于云数据库的动态性、可伸缩性、大规模性有很好的契合点^[3]。

4 蚁群算法基本原理及全局规划算法

4.1 蚁群算法的基本原理

经研究发现:蚂蚁在觅食过程中能够在所经过的路径上留下一称为信息素的物质,而且蚂蚁能够感知这种物质的存在及其强度,并以此指导自己的运动方向,它们会朝着该物质强度高的方向移动^[4]。因此,由大量蚂蚁组成的集体觅食行为表现出:某一路径越短,该路径上走过的蚂蚁就越多,则留下的信息素强度就越大,后来的蚂蚁选择该路径的概率就越大^[5]。蚂蚁个体之间就是通过这种信息交流来选择最短路径并达到搜索食物的目的。

4.2 全局规划算法

为了叙述方便,进一步做出如下约定和定义。

定义 1 $ant_i = \{1, 2, \dots, k, \dots, m\}$ 表示一个蚂蚁家族所有蚂蚁的集合($i=1, 2$), $k \in ant_i$ 表示某只蚂蚁, m 为第 i 蚂蚁家族的蚂蚁总数, $\tau_{ij}(t)$ 表示蚂蚁在 t 时刻残留在栅格 g_i, g_j 连线上的信息量。

定义 2 $\forall g \in A, g \notin OS$, 则称 g 为可行节点,所有可行节点的集合称为可行域,记为 FS ; $\forall g \in A, g \in OS$, 则称 g 为禁入节点,所有禁入节点的集合称为禁入域,记为 NFS 。显然, $FS = A \cap (OS)^c$; $NFS = A \cap OS$; 其中,上标 c 表示补集。

算法步骤:

考察蚂蚁在云数据库中的路径规划问题,若 ANT 在时刻 t_i 处于 g_i 位置,它下一步要走的栅格范围必然是 g_i 邻域 BR_i 内的可行域中的可行点。因此,蚂蚁从当前节点 g_i 选择下一节点时,只要在 $Wk_i(g_i)$ 内选择 g_j 即可。算法步骤如下:

(1)初始化:将 ANT 放置在出发点 g_{begin} ,并设置到禁忌表 $tabu_k$ 中。令 $\tau_{ij}(0) = \tau_0$ (τ_0 为常数)。设置觅食代数计数器 $n=0$,最大代数为 max 。

(2) $\forall k$,以当前节点 $g_i \in FS$ 为中心, $i \in C$,按最近临近选择策略选择并行走走到下一个节点 g_j ,且有 $g_j \in Wk_i(g_i)$, $g_j \notin tabu_k$,节点则根据与其相邻结点的概率值的大小进行选择,选择概率值最大的结点来作为蚂蚁的下一个结点。

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)}{\sum_{k \in allowed_k} \tau_{ij}^{\alpha}(t)}, & j \in allowed(i) \\ 0, & j \notin allowed(i) \end{cases} \quad (1)$$

式中, $j \in C$, 蚂蚁 k 所选 g_j 的节点序号; S 为由式(1)所决定

的随机变量; q 为随机数; q_0 为初始化时所给的阈值; $\eta_i(g_i)$ 为由前面所定义给出的启发信息; β 为在边上残留信息的重要程度; $P_{ij}^k(n)$ 在 n 代,蚂蚁 k 由节点 i 转移到节点 j 的概率, $i, j \in C$ 。

q 和 q_0 是为了防止出现停滞而设的随机搜索策略所需参数,以增加搜索的多样性。当 $q > q_0$ 时,计算 $|Z|$ 个节点的转移概率,并根据赌轮盘规则选择节点 j ,然后将 j 加入禁忌表 $tabu_k$ 。

(3) 局部信息素更新

随着时间的推移,以前留下的信息素逐渐消失,用参数 $1-\rho$ 表示信息消失程度,每一只蚂蚁选择完一个节点即走完一个边后,按下式进行局部信息更新。

$$\tau_{ij}(n+1) = (1-\rho)\tau_{ij}(n) + \rho\Delta\tau_{ij}^k \quad (2)$$

式中,

$$\rho\Delta\tau_{ij}^k = \begin{cases} \frac{Q_k}{l_{ij}^k}, & \text{当蚂蚁 } k \text{ 走过边 } e_{ij} \\ 0, & \text{否则} \end{cases} \quad (3)$$

$$l_{ij}^k = \sum_{l=1}^m dl \quad (4)$$

$$\text{当 } \tau_{ij}(n+1) < \tau_{\min} \text{ 时, } \tau_{ij}(n+1) = \tau_{\min} \quad (5)$$

式中, Q_k 为常数, $\tau_{\min}$ 是设定的最小值; d_l 是蚂蚁 k 已走过的边的边长。

(4) 全局信息素更新

本次觅食相遇并完成通道连接后,将本次觅食最短通道上的信息素按下式调整:

$$\tau_{ij}^{new} = (1-\alpha)\tau_{ij}^{old} + \alpha\Delta\tau_{ij} \quad (6)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (7)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q_0}{L_{k\min}}, & ij \in \phi \\ 0, & \text{其它} \end{cases} \quad (8)$$

式中, Q_0 为常数; α 为全局信息素挥发系数; $L_{k\min}$ 为本次觅食最佳通道的路径; ϕ 表示蚂蚁 k 所走的边 ij 属于最佳。

(5) $\forall k, k=1, 2, \dots, m$ 选择第 j 个节点后,按前面的定义的条件,检查蚂蚁是否与食物相遇,若有,则转步骤(6);否则,令 $i=j$ (即以新选的节点 j 作为当前节点),返回步骤2开始选择下一个节点,直到找到食物或所有节点选择完毕。

(6) 如果找到,则蚂蚁按照原来路径逆向返回源结点。

5 仿真实验

5.1 实验模型

一个数据网络的作用取决于许多以非线性和不可预测的方式相互作用的组成部分,所以选择一个有意义的测试环境是非常困难的:遵循的方法就是定义一个可调成分的各种分类组成的有限集合^[6]。该实验模型选择 South California University ISI 研究院的 NS-2 作为该实验平台^[7]。同时把吞吐量(每秒内正确传输的比特数)和数据分组的发送延迟作为对性能评价的度量标准;并且选择了在通信和机器学习中的优秀的路由算法 OSPF(Open Shortest Path First)、SPF(Shortest Path First)以及 BF 来与该算法进行对比。实验的数据是 20 次实验的平均值,每次实验的仿真时间为 1000 个虚拟秒。

5.2 实验结果

笔者研究该算法在增加流量负载以及在暂时饱和的情况

下具有的性能。

从图 1 的实验结果来看,该算法(IANT)在平均吞吐量性能要高于其它 3 个算法,而其它的 3 个算法差异不是很大,对于算法(IANT)来说,由于其参数的值从 2.4 逐次降到 2.0 时,其吞吐量会有相应的降低。

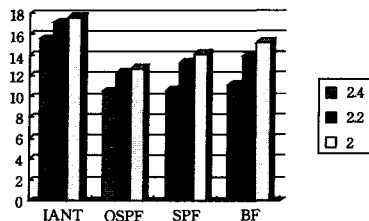


图1 平均吞吐量

从图 2 的实验结果来看,该算法 IANT 的分组路由的时间延迟性能要好于其它 3 个算法。其中 BF 算法的分组时间延迟性能接近于 IANT 算法,而 OSPF 和 SPF 的性能接近。

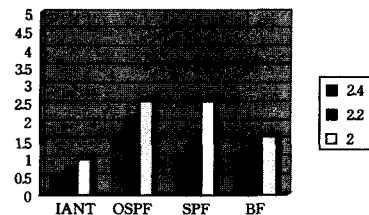


图2 分组延迟的经验分布

5.3 实验分析

通过实验,可以发现:

α :启发因子, α 太小,收敛速度太慢,易陷入局部最优; α 太大,信息素在搜索过程中的权重大,会出现过早收敛。

β :期望启发式因子,与启发因子关联性很强。 β 太小将导致蚂蚁群体陷入纯粹的随机搜索,在这种情况下很难找到最优解; β 太大时,收敛速度加快,但其收敛性能有变差的趋势。

ρ :信息素挥发因子。 ρ 过大时,以前搜索过的路径被再次选择的可能性过大,会影响算法的随机性和全局搜索能力;反之,可以提高算法的随机性能和全局搜索能力,但会使算法的收敛速度降低。

Q :信息素增强因子。 Q 越大,则在蚂蚁已遍历路径上信息素的积累加快,可以增强蚁群搜索时的正反馈能力,有助于算法的快速收敛;但当 Q 过大时,算法的全局搜索能力变差,易陷入局部最优解,导致环路的出现,也会造成网络性能评估的大幅度波动。

结束语 该文把蚁群智能搜索算法和当前流行的云数据库的搜索结合起来,并在传统的蚁群算法的基础上进行改进,运用了吞吐量和传输时间延迟作为衡量算法性能的两个指标,并利用功能优越的 NS-2 网络实验平台,对云数据库进行查询,并把结果与其它的路由算法进行对比,得到了算法参数是如何影响算法性能的结论,得到了较为良好的实验结果和效果。下一步的研究工作是如何优化搜索性能,减少搜索时间,降低搜索成本。

参考文献

[1] <http://soft.ccw.com.cn/it/>

(下转第 213 页)

$$O(2^{2k}kz) \sum_{q=1}^z n_q = O(2^{2k}knz)。$$

3.3 整个枚举算法

对于 3-维匹配问题的任意一个实例 S , 给出枚举 S 中权值最大的 z 个 k -matchings 的算法。图 2 详细描述了该算法的主要步骤。

算法 EMM(S, k, z)

输入: 一个含有 n 个带权的元组集合 S , 两个非负整数 k, z

输出: S 中权值最大的 z 个 k -matchings

```

1  $R = \emptyset$ ;
2 对  $S$  的第 2, 3 列符号构造一个  $(2k)$ -着色方案  $H$ ;
3 for  $H$  中的每一个  $(2k)$ -着色函数  $f_i$  do
    3.1  $T = \text{ELMM}(S, k, z, f_i)$ ;
    3.2  $R = R \cup T$ ;
4 if  $R \neq \emptyset$ 
    4.1 then {用基数排序法删除  $R$  中重复的  $k$ -matching;
    4.2 在  $R$  中输出权值最大的  $z$  个  $k$ -matchings;}
4.3 else 提示不存

```

图 2 3-维匹配问题的固定参数枚举算法

定理 2 设 S 为带权的 3-维匹配问题的一个含有 n 个元组的实例, 则算法 EMM(S, k, z) 能在时间 $O(5 \cdot 48^{3k} kn^2 z)$ 内枚举 S 中权值最大的 z 个 k -matchings。

证明: 首先证明算法 EMM(S, k, z) 的正确性。设 ma 为 S 中的任意一个 k -matching, H 为 S 中第 2, 3 列符号的一种 $(2k)$ -着色方案, 由着色方案的定义可知, ma 一定被 H 中的某一个 f_i 正确着色。进一步地, 如果 ma 为 S 中权值最大的 z 个 k -matchings 中的任意一个, 则由算法 ELMM(S, k, z, f_i) 的正确性可知, ma 一定由算法 ELMM(S, k, z, f_i) 输出, 即一定包含在集合 R 中, 因此在 R 中选择权值最大的 z 个 k -matchings 就是实例 S 的权值最大的 z 个 k -matchings。如果 S 中不存在 k -matching, 算法的第 4 步将返回空集。

为避免集合 R 中出现 k 个元组完全相同的多个 k -matchings, 算法的第 4.1 步用基数排序删除可能重复的 k -matching。基数排序时用 k -matching 中 k 个元组名的组合作为关键字, 元组名顺序均按实例 S 中元组的自然顺序排列。

接下来分析它的时间复杂度。算法 EMM(S, k, z) 第 2 步的运行时间也就是着色算法的运行时间, 即 $O(6 \cdot 4^{2k} n)$, 第 3 步用 $O(6 \cdot 4^{2k} n)$ 个着色函数依次对 S 的第 2, 3 列符号进行着色, 并求出每种着色函数映射下的权值最大的 z 个 k -matchings, 运行时间为 $O(2^{2k} nkz) \times O(6 \cdot 4^{2k} n) = O(12 \cdot 8^{2k} n^2 kz)$, 产生最多 $O(6 \cdot 4^{2k} nz)$ 个权值最大的 k -matchings 并构成集合 R 。第 4.1 步基数排序需运行时间 $O(6 \cdot 4^{2k} nzk)$ 。第 4.2 步对已清除重复 k -matching 的 R 可以在 $O(6 \cdot 4^{2k} nz)$ 时间内找到第 z 大的 k -matchings, 然后在 $O(6 \cdot 4^{2k} nz)$ 时间内找出权值最大的 z 个互不相同的 k -matchings。

因此, 算法 EMM(S, k, z) 的时间复杂度为: $O(6 \cdot 4^{2k} n) +$

$$O(12 \cdot 8^{2k} n^2 kz) + O(6 \cdot 4^{2k} nkz) + O(6 \cdot 4^{2k} nz) = O(12 \cdot 8^{2k} n^2 kz) = O(5 \cdot 48^{3k} kn^2 z)。$$

定理 2 表明产生 z 个权值最大的 k -matchings 的“平均”时间是 $O(5 \cdot 48^{3k} kn^2 z)$ 。

推论 1 3-维匹配问题是固定参数线性可枚举的。

结束语 本文运用最新的着色技术和动态规划技术对带权的 3-维匹配问题提出了一个时间复杂度为 $O(5 \cdot 48^{3k} kn^2 z)$ 的固定参数枚举算法, 表明了带权的 3-维匹配问题是固定参数线性可枚举的。

本文的结论具有重要的理论意义。对于 3-维匹配问题的任意一个实例, 枚举其中所有的 k -matchings 目前还不存在高效算法, 但本文的结论表明枚举其中权值最大的 z 个 k -matchings 一定存在一个高效算法。从这个意义上讲, 3-维匹配问题是又一个固定参数可枚举的典型例子。

3-维匹配问题有着广泛的应用背景, 如线路布置、多级调度和工种分派等。因此本文提出的固定参数枚举算法也具有较大的应用价值。

参考文献

- [1] Valiant L. The complexity of enumeration and reliability problems [J]. SIAM Journal on Computing, 1979, 8(3): 410-421
- [2] Matsui Y, Matsui T. <http://dmawww.epfl.ch/roso.mosaic/kf/enum/comb/combenum.html>
- [3] Flum J, Grohe M. The parameterized complexity of counting problems [J]. SIAM Journal on Computing, 2004, 33(4): 892-922
- [4] Chen J, Kanj I, Meng J, et al. On effective enumerability of NP problems [C] // Proceedings of the 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 215-226
- [5] Liu Y, Lu S, Chen J, et al. Greedy localization and color-coding: improved matching and packing algorithms [C] // Proceedings of the 2nd International Workshop on Parameterized and Exact Computation. Berlin: Springer-Verlag, 2006: 84-95
- [6] 宁丹, 王建新. 一个基于着色和动态规划的 3-维匹配问题算法 [J]. 计算机科学, 2007, 34(7): 222-224
- [7] Liu Y, Chen J, Wang J. On counting 3-D matchings of size k [J]. Algorithmica, 2009, 54: 349-359
- [8] Chen J, Lu S, Sze S-H, et al. Improved algorithms for path, matching, and packing problems [C] // Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. 2007: 298-307
- [9] Alon N, Yuster R, Zwick U. Color-coding [J]. Journal of the ACM, 1995, 42: 844-856
- [10] Cormen T, Leiserson C, Rivest R, et al. Introduction to Algorithms 2nd Edition [M]. Boston, MA: McGraw-Hill Book Company, 2001

(上接第 145 页)

- [2] www.aco-metaheuristic.org
- [3] Colomni A, Dorigo M, Maniezzo V. Distributed Optimization by ant colonies [C] // F. J. Varela & P. Bourgin, eds. Proceedings of the First European Conference on Artificial Life. Cambridge, MA, MIT Press, 134-412
- [4] Minton S, Johnston. Minimizing conflicts: A heuristic repair me-

thod for constraint satisfaction and scheduling problems [J]. Artificial Intelligence, 1992

- [5] Reischle M F, Schmeck H. Multi colony ant algorithms [J]. Journal of Heuristics, 2002
- [6] Dorigo M, Stutzle T. Ant Colony Optimization. 2007
- [7] www.isi.edu/nsnam/ns/